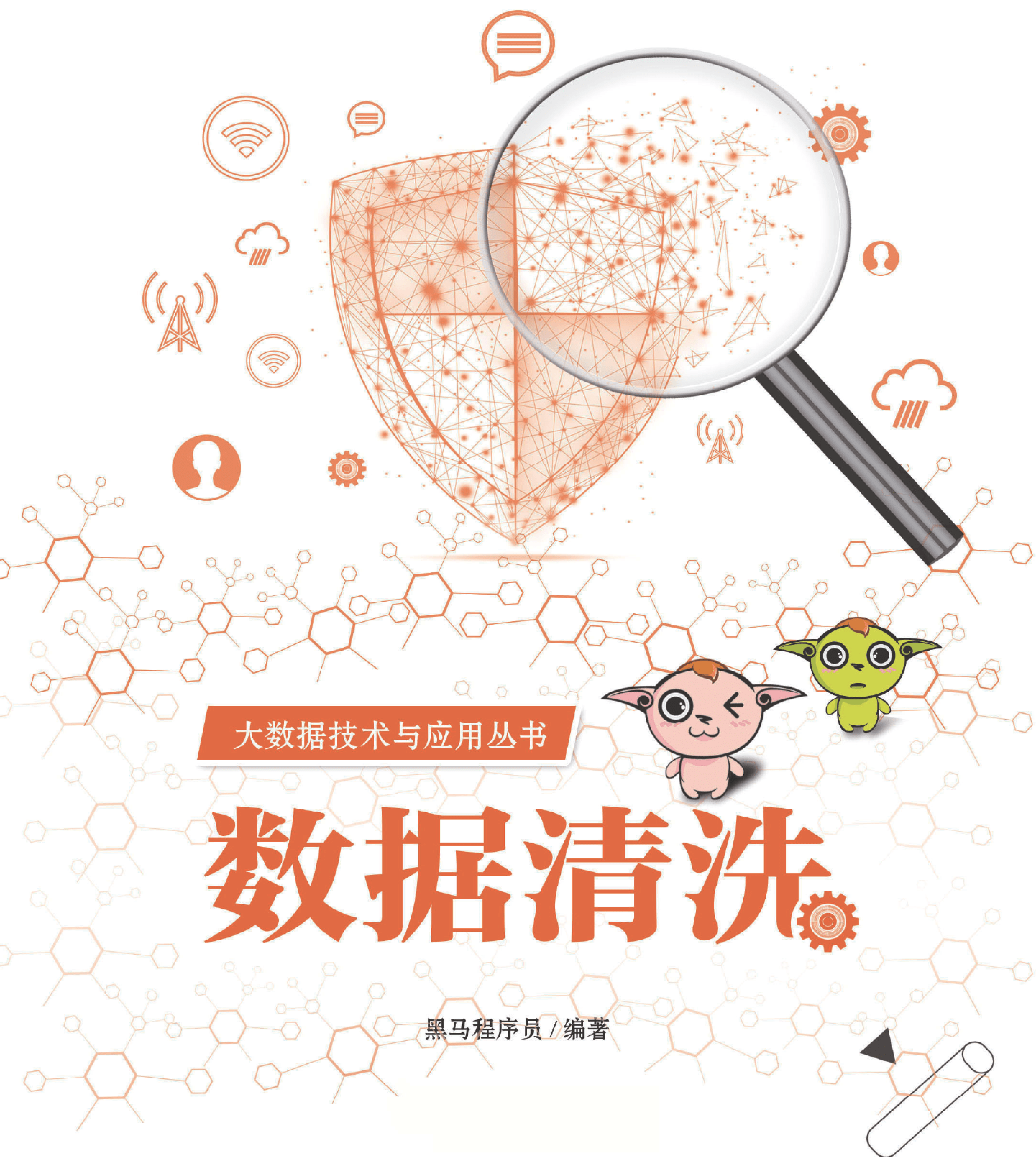
NITE 国家软件与集成电路公共服务平台信息技术紧缺人才培养工程指定教材



大数据技术与应用丛书

数据清洗

黑马程序员 / 编著

清华大学出版社

大数据技术与应用丛书

数据清洗

黑马程序员　编著

清华大学出版社
北　京

内 容 简 介

数据清洗是大数据预处理的关键环节。面对错综复杂的数据，传统的清洗“脏”数据工作单调且异常辛苦，如果能利用正确的工具和方法，可以让数据清洗工作变得事半功倍。本书讲解数据清洗的理论知识和实际应用，全书共8章：第1章主要带领大家简单认识数据清洗；第2章主要讲解 ETL 技术相关的知识；第3章讲解 Kettle 工具的基本使用；第4章主要讲解数据清洗的第一步——数据抽取；第5章主要讲解数据清洗与检验；第6章主要讲解数据转换相关的知识；第7章主要讲解数据加载相关的知识；第8章利用前面章节所学的知识构建一个 DVD 租赁商店数据仓库，目的是实现定期从源数据库 sakila 中抽取增量数据，转换成符合 DVD 租赁业务的数据，最后加载到 DVD 租赁商店数据仓库中，便于后续在线 DVD 租赁商店的决策者对数据进行分析得出商业决策。本书附有配套视频、源代码、习题、教学设计、教学课件等资源。同时，为了帮助初学者更好地学习本书中的内容，还提供了在线答疑，欢迎读者关注。

本书可作为高等院校本专科计算机、信息管理等相关专业的大数据课程教材，也可供相关技术人员参考，是一本适合广大计算机编程爱好者的优秀读物。

图书在版编目(CIP)数据

数据清洗/黑马程序员编著. —北京：清华大学出版社，2020.3
(大数据技术与应用丛书)
ISBN 978-7-302-55087-7

Ⅰ. ①数…　Ⅱ. ①黑…　Ⅲ. ①数据处理　Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字(2020)第 044515 号

责任编辑： 袁勤勇　常建丽
封面设计： 韩　冬
责任校对： 徐俊伟
责任印制： 丛怀宇

出版发行： 清华大学出版社
网　　址： http://www.tup.com.cn，http://www.wqbook.com
地　　址： 北京清华大学学研大厦 A 座　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载： http://www.tup.com.cn，010-83470236
印 装 者： 三河市龙大印装有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 25.5　　**字　　数：** 635 千字
版　　次： 2020 年 4 月第 1 版　　**印　　次：** 2020 年 4 月第1 次印刷
定　　价： 49.80 元

产品编号：086727-01

序言

江苏传智播客教育科技股份有限公司(简称"传智播客")是一家致力于培养高素质软件开发人才的科技公司。经过多年探索,传智播客的战略逐步完善,从IT教育培训发展到高等教育,从根本上解决以"人"为单位的系统教育培训问题,实现新的系统教育形态,构建出前后衔接、相互呼应的分层次教育培训模式。

一、"黑马程序员"——高端IT教育品牌

"黑马程序员"的学员多为大学毕业后,想从事IT行业,但各方面条件还不成熟的年轻人。"黑马程序员"的学员筛选制度非常严格,包括了严格的技术测试、自学能力测试,以及性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保学员质量,并降低企业的用人风险。

自"黑马程序员"成立以来,教学研发团队一直致力于打造精品课程资源,不断在产、学、研3个层面创新自己的执教理念与教学方针,并集中"黑马程序员"的优势力量,有针对性地出版了计算机系列教材90多种,制作教学视频数十套,发表各类技术文章数百篇。

"黑马程序员"不仅斥资研发IT系列教材,还为高校师生提供以下配套学习资源与服务。

1. 为大学生提供的配套服务

(1) 请同学们登录 http://yx.ityxb.com,进入"高校学习平台",免费获取海量学习资源。平台可以帮助高校学生解决各类学习问题。

(2) 针对高校学生在学习过程中存在的压力大等问题,我们还面向大学生量身打造了IT技术女神——"播妞学姐",可提供教材配套源码、习题答案及更多学习资源。同学们快来关注"播妞学姐"的微信公众号 boniu1024。



"播妞学姐"微信公众号

2. 为教师提供的配套服务

针对高校教学,"黑马程序员"为IT系列教材精心设计了"教案+授课资源+考试系统+题库+教学辅助案例"的系列教学资源。高校老师请登录 http://yx.ityxb.com,进入"高校教辅平台",也可关注"码大牛"老师微信/QQ:2011168841,获取配套资源,还可以扫

描下方二维码，关注专为 IT 教师打造的师资服务平台——“教学好助手”，获取最新的教学辅助资源。



“教学好助手”微信公众号

二、“传智专修学院”——高等教育机构

传智专修学院是一所由江苏省宿迁市教育局批准、江苏传智播客教育科技股份有限公司投资创办的四年制应用型院校。学校致力于为互联网、智能制造等新兴行业培养高精尖科技人才，聚焦人工智能、大数据、机器人、物联网等前沿技术，开设软件工程专业，招收的学生入校后将接受系统化培养，毕业时学生的专业水平和技术能力可满足大型互联网企业的用人要求。

传智专修学院借鉴卡内基·梅隆大学、斯坦福大学等世界著名大学的办学模式，采用“申请入学，自主选拔”的招生方式，通过深入调研企业需求，以校企合作、专业共建等方式构建专业的课程体系。传智专修学院拥有顶级的教研团队、完善的班级管理体系、匠人精神的现代学徒制和敢为人先的质保服务。

传智专修学院突出的办学特色如下。

（1）立足“高精尖”人才培养。传智专修学院以国家重大战略和国际科学技术前沿为导向，致力于为社会培养具有创新精神和实践能力的应用型人才。

（2）项目式教学，培养学生自主学习能力。传智专修学院打破传统高校理论式教学模式，将项目实战式教学模式融入课堂，通过分组实战，模拟企业项目开发过程，让学生拥有真实的工作能力，并持续培养学生的自主学习能力。

（3）创新模式，就业无忧。学校为学生提供“一年工作式学习”，学生能够进入企业边工作边学习。与此同时，我们还提供专业老师指导学生参加企业面试，并且开设了技术服务窗口给学生解答工作中遇到的各种问题，帮助学生顺利就业。

如果想了解传智专修学院更多的精彩内容，请关注微信公众号“传智专修学院”。



传智专修学院

传智播客

2020 年 2 月

前言

近年来，大数据技术掀起了计算机领域的一个新浪潮，无论是数据挖掘、数据分析、数据可视化，还是机器学习、人工智能，它们都绕不开“数据”这个主题。从统计学家到软件开发人员，再到图形设计师，越来越多的人对数据科学产生了兴趣，廉价的硬件、可靠的数据处理工具和数据可视化工具以及海量的数据，这些资源使我们能够轻松地、精确地发现趋势、预测未来。

由于海量数据的来源是广泛的，数据类型也是多而繁杂的，因此数据中会夹杂着不完整的、重复的以及错误的数据，如果直接使用这些原始数据，会严重影响数据决策的效率。因此，对原始数据进行有效的清洗是大数据分析和应用过程中的关键环节。

本书共分为8章，各章内容介绍如下：

第1章主要是带领大家简单认识数据清洗的背景、定义、原理、基本流程、清洗策略以及常见的数据清洗方法。通过本章的学习，读者可以对数据清洗有基本的认识，便于后续章节的学习。

第2章主要讲解ETL技术相关的知识，即基于ETL的数据清洗(ETL的概念、体系结构)、ETL关键技术(抽取数据的技术、数据清洗转换的技术以及数据加载的技术)和ETL常见工具。通过本章的学习，读者可以认识ETL，并熟悉ETL的关键技术和ETL常见的工具。

第3章主要讲解数据清洗工具Kettle的相关知识，即Kettle的概述、设计原则、下载安装以及基本概念和基本功能。通过本章的学习，读者可以使用Kettle工具对ETL数据进行相关处理(抽取、清洗转换以及加载)。

第4章主要讲解数据清洗的第一步——数据抽取，即抽取文本数据、Web数据以及数据库数据的相关知识。通过本章的学习，读者可以掌握抽取各种形式的数据并保存至数据库中，便于后续对数据进行相关清洗转换和分析。

第5章主要讲解数据的清洗与检验相关的知识，即数据去重、缺失值处理、异常值处理以及数据检验知识。通过本章的学习，读者可以掌握对重复数据、缺失值数据、异常值数据的处理，也可以掌握对数据进行检验的操作。

第6章主要讲解数据转换相关的知识，即多数据源合并、不一致数据转换、数据粒度的转换、数据的商务规则计算的知识。通过本章的学习，读者可以掌握数据转换操作，实现将企业中的数据进行规范化处理。

第7章主要讲解数据加载相关的知识，即数据的加载机制(全量加载和增量加载)和批量加载的知识。通过本章的学习，读者可以掌握数据的全量加载、增量加载以及批量加载的操作，实现将清洗检验、转换后的高质量企业数据加载到目标数据库中，便于后续进行数据

分析和数据挖掘。

第8章利用前面章节所学的知识构建一个DVD租赁商店数据仓库，目的是实现定期从源数据库sakila中抽取增量数据，转换成符合DVD租赁业务的数据，最后加载到DVD租赁商店数据仓库中，便于后续在线DVD租赁商店的决策者对数据进行分析得出商业决策。通过本章的学习，读者可以掌握数据仓库的构建操作以及对数据库中的数据进行相关清洗转换操作。

致谢

本书的编写和整理工作由传智播客教育科技股份有限公司完成，主要参与人员有高美云、文燕、张明强等，全体参编人员在编写过程中付出了许多辛勤的汗水。除此之外，传智播客等600多名学员也参与了本书的试读工作，他们站在初学者的角度对本书提供了许多宝贵的意见，在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力，但书中难免会有不妥之处，欢迎各界专家和读者朋友提出宝贵意见。您在阅读本书时，如果发现任何问题或有不认同之处，可以通过电子邮件与我们取得联系。请发送电子邮件至：itcast_book@vip.sina.com。

黑马程序员

2019-09-23于北京

目录

第1章 数据清洗概述

学习目标

(1) 了解数据清洗的背景
(2) 了解数据清洗的定义
(3) 熟悉数据清洗的原理
(4) 掌握数据清洗的基本流程
(5) 了解常见数据清洗的策略和方法

近年来,大数据技术掀起了计算机领域的一个新浪潮,无论是数据挖掘、数据分析、数据可视化,还是机器学习、人工智能,它们都绕不开"数据"这个主题。从统计学家到软件开发人员,再到图形设计师,越来越多的人对数据科学产生了兴趣。廉价的硬件、可靠的数据处理工具和数据可视化工具以及海量的数据这些资源使我们能够轻松地、精确地发现趋势、预测未来。

由于海量数据的来源是广泛的,数据类型也是多而繁杂的,因此数据中会夹杂着不完整、重复以及错误的数据,如果直接使用这些原始数据,会严重影响数据决策的准确性和效率。因此,对原始数据进行有效的清洗是大数据分析和应用过程中的关键环节。本章将针对数据清洗的相关知识进行详细讲解。

1.1 数据清洗的背景

当今时代,企业信息化的要求越来越迫切。对于企业的决策者来说,正所谓"垃圾进垃圾出(garbage in,garbage out)"——如果作为决策支持的数据仓库存放的数据质量达不到要求,将直接导致数据分析和数据挖掘不能产生理想的结果,甚至还会产生错误的分析结果,从而误导决策。因此,我们需要对数据仓库中的数据进行相关清洗操作,得出可靠、可准确反映企业实际情况的数据,用以支持企业战略决策。由此可见,数据质量在企业战略决策中占据着重要的地位。本节将讲解数据质量概述、数据质量的评价指标以及数据质量的问题分类。

1.1.1 数据质量概述

数据质量是指在业务环境下,数据符合数据消费者的使用目的,能满足业务场景具体需求的程度。但是,在不同的业务场景中,数据消费者对数据质量有各自不同的观点,具体如下:

- 对于一个邮件列表的管理员来说，数据质量与姓名、地址有关，高质量的数据意味着清晰、准确、不存在二义性以及不重复的邮件传送地址。
- 对于数据清洗工具销售商来说，数据质量与姓名、地址有关，以及与它们的工具是否能够规范地校验和匹配客户记录有关。
- 对于数据仓库工程师来说，数据质量是将他们接收的应用数据经过相关的处理，存储到表格中或者显示到窗口中。
- 对于一个数据挖掘和决策支持系统的使用者来说，数据质量意味着准确、无重复且符合许多特定要求的数据。

从适用性的角度看，数据质量是一个相对的概念（与决策有关）。不同的决策者对数据质量的高低要求也是不同的。对于一个无关的数据，即使质量很高，对决策也起不到任何作用。例如，医院里病人的基本信息通常包括姓名、年龄、血型、身高、地址等内容，如果想研究某种疾病易发的年龄段，那么年龄信息的数据质量就非常重要，而其他信息（血型、身高、地址等）的数据质量相对来说作用不大。

数据质量的显著特点如下。

- “业务需求”会随时间变化，数据质量也会随时间发生变化。
- 数据质量可以借助信息系统度量，但独立于信息系统存在。
- 数据质量存在于数据的整个生命周期，随着数据的产生而产生，随着数据的消失而消失。

1.1.2 数据质量的评价指标

数据质量的评价指标主要包括数据的准确性（accuracy）、完整性（completeness）、简洁性（concision）及适用性（applicability），其中数据的准确性、完整性和简洁性是为了保证数据的适用性。下面针对数据质量的主要评价指标进行详细的介绍。

1. 准确性

数据的准确性就是要求数据中的噪声尽可能少。为提高数据的准确性，需对数据集进行降噪处理。对于数据中偏离常规、分散的小样本数据，一般可视为噪声或异常数据，可通过最常用的异常值检测方法聚类进行处理。

2. 完整性

完整性指的是数据信息是否存在缺失的状况。数据缺失的情况可能是整条数据记录缺失，也可能是数据中某个字段信息的记录缺失。不完整的数据所能借鉴的价值会大大降低，也是数据质量更为基础的一项评估标准。

数据质量的完整性比较容易评估，一般通过数据统计中的记录值和唯一值进行评估。例如，网站日志日访问量就是一个记录值，平时的日访问量在1000左右，突然某天降到100，就需要检查数据是否存在缺失了。

3. 简洁性

简洁性就是要尽量选择重要的本质属性，并消除冗余。进行决策时，决策者往往抓住反

映问题的主要因素，而不需要把问题的细节都搞得很清楚。在数据挖掘时，特征的个数越多，产生噪声的机会就越大。一些不必要的属性既会增大数据量，又会影响挖掘数据的质量。因此，选择较小的典型特征集不仅符合决策者的心理，而且还容易挖掘到简洁有价值的信息。

4. 适用性

适用性是评价数据质量的重要标准。建立数据仓库的目的是进行数据挖掘、支持决策分析，而在现实世界中很难挖掘到满意的数据，但是我们可以尽量获取符合要求的数据。数据的质量是否能满足决策的需要是适用性的关键所在。尽管前面已经强调了数据的准确性、完整性和简洁性，但归根结底是为了数据的实际效用。从数据的实际效用上讲，适用性才是评价数据质量的核心准则。

1.1.3 数据质量的问题分类

数据质量的问题可以分为两类：一类是基于数据源的"脏"数据分类；另一类是基于清洗方式的"脏"数据分类。下面分别针对基于数据源的"脏"数据分类和基于清洗方式的"脏"数据分类进行详细讲解。

1. 基于数据源的"脏"数据分类

通常情况下，将数据源中不完整、重复以及错误等有问题的数据称为"脏"数据。由于数据仓库的数据来自底层数据源，因此"脏"数据出现的原因与数据源有密切的关系。基于数据源的"脏"数据分类如图 1-1 所示。

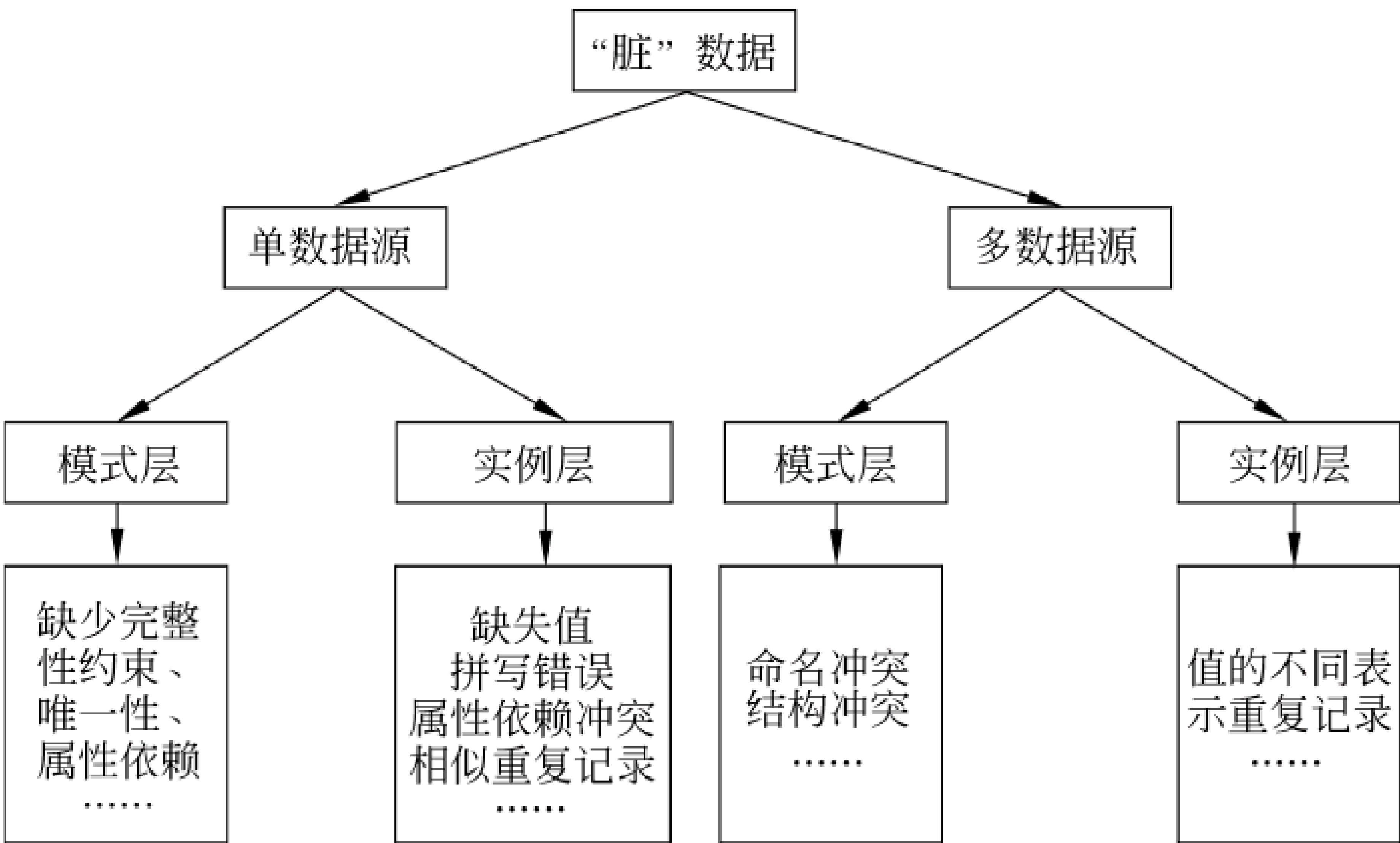


图 1-1 基于数据源的"脏"数据分类

从图 1-1 中可以看出，基于数据源的"脏"数据质量问题可以分为两类，即单数据源问题和多数据源问题。单数据源问题和多数据源问题的具体介绍如下。

1）单数据源问题

单数据源的数据质量主要取决于它的模式对数据完整性约束的控制程度。由于数据模式和完整性约束控制了数据的范围，如果单数据源没有数据模式，就会对进入和存储的数据

缺乏相应的限制，此时很有可能出现拼写错误的数据和不一致的数据。

单数据源的实例层问题是由于数据在模式层无法预防的错误和不一致引起的。典型的单数据源实例层问题包括缺失值(即一些记录在某些属性上没有值)、拼写错误(即在数据输入时容易出现)、属性依赖冲突(即不满足属性间的依赖关系，如城市名与邮政编码不满足对应关系等)以及相似重复记录(即由于数据输入错误等原因导致有多条记录表示现实世界中的同一个实体)。

对于不同范围的数据质量问题，相应的数据清洗方式也会有所不同，清楚地了解目标数据存在的质量问题是提供完善的数据清洗方式的基础。

2）多数据源问题

单数据源情况下出现的问题在多数据源情况下变得更加严重。每个数据源中都有可能包含"脏"数据，而且每个数据源中的数据表示方法都各自不同，还有可能出现数据重复或矛盾冲突。因为在很多情况下，各个数据源都是为了满足某一个特定需要而单独设计、配置和维护，这很大程度上导致数据库管理系统、数据模型、模式设计和实际数据的异构性。

多数据源中存在的与模式相关的质量问题主要是名字冲突和结构冲突。名字冲突表现在同一个名字表示不同的对象，或不同的名字表示同一个对象；结构冲突的典型表现是不同的数据源中同一对象用不同的方式表示。

除模式相关的质量问题外，许多质量问题只出现在实例层次上。单数据源中出现的各种问题都将以不同方式出现在不同的数据源中，如重复记录、矛盾记录等。即使在具有相同属性名称和数据类型的情况下，各异构数据源中的数据也可能有不同的表示方式，或不同的解释在不同的数据源中信息的聚集程度以及代表的时间点都有可能不同。

2. 基于清洗方式的"脏"数据分类

基于数据源的"脏"数据分类方法需要为每种类型的"脏"数据设计单独的清洗方式。从数据清洗方式的设计者角度看，可以将"脏"数据分为"独立型"脏"数据"和"依赖型"脏"数据"两类。基于清洗方式的"脏"数据分类如图 1-2 所示。

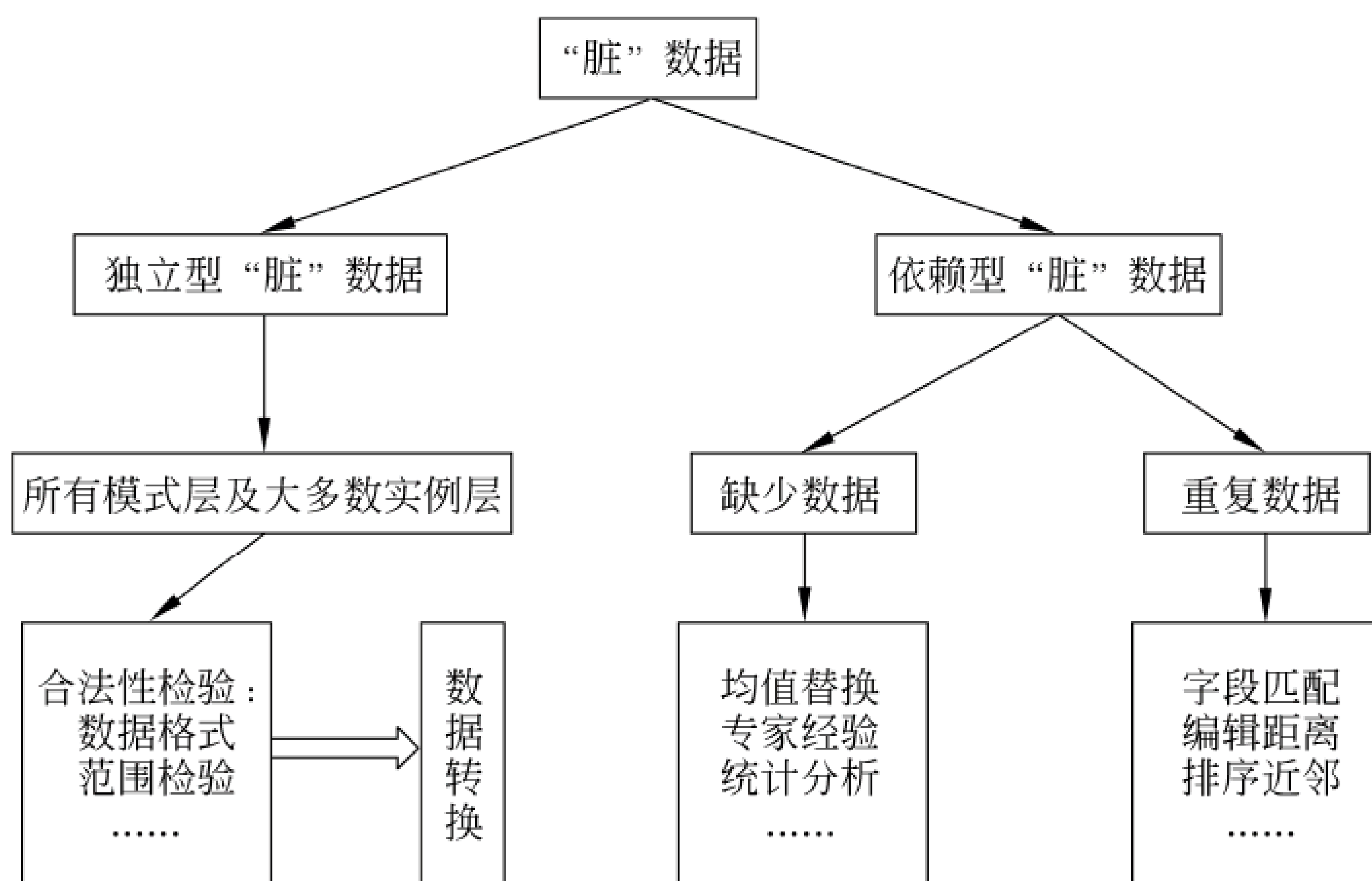


图 1-2　基于清洗方式的"脏"数据分类

从图 1-2 中可以看出，独立型“脏”数据包括单数据源和多数据源所有模式层及大多数实例层的数据质量问题；依赖型“脏”数据包括缺失数据和重复数据等“脏”数据。下面对独立型“脏”数据和依赖型“脏”数据进行详细讲解。

1）独立型“脏”数据

独立型“脏”数据可通过记录或本身属性检验出是否包含“脏”数据，不需要依赖其他记录或属性检测。独立型“脏”数据使用“数据合法性检验规则”检测数据字段的实际内容，若属性值不符合规则，则视为“脏”数据，此时可调用已定义的相关清洗方式，将其变为满足规则的数据，从而可以保证数据的合法性。

合法性检验是判断数据是否符合给定标准的过程，判断标准是用户根据业务需要定义的一些检验规则，该规则主要检验的是数据的格式、数据的范围、数据的枚举清单以及数据的相关性等方面，具体介绍如下。

- 数据的格式主要是检验记录的某个字段或字段组中的数据是否符合规范格式，这是针对模式层的“脏”数据进行检验。
- 数据的范围主要是检查记录的字段数据是否在预期的范围内，常用于检验数字和有效值。
- 数据的枚举清单主要是参照某个已定义的清单检验字段的值。
- 数据的相关性主要通过主键和外键的关系实现。

综上所述，数据的合法性检验是一个非常耗时的环节，但也是一个必不可少的环节，因此，该环节应高度自动化。在设计清洗程序时，应该内置较多的检验函数和环节，这样可以减少用户定制数据合法性检验规则的工作量。

数据转换是将“脏”数据进行清洗的过程，包括模式转换和实例转换。其中，模式转换用来解决模式层“脏”数据的问题，通过在元数据库中定义表字段的映射规则、字段拆分规则以及字段值合并规则等协调数据模式之间的差异，从而实现数据的清洗；实例转换是根据源数据字段的实际内容，结合一定的领域知识解决拼写错误、输入错误、不同的计量单位及过时的编码等实例层“脏”数据问题。

2）依赖型“脏”数据

依赖型“脏”数据主要包括缺失数据和重复数据等“脏”数据。由于需要综合考虑与其他记录间的关联，依赖型“脏”数据的处理很难有通用的方法。一般地，针对特定类型的“脏”数据设计特定的清洗方式。

（1）缺失数据。

缺失数据主要包括数据空值和数据异常，具体介绍如下。

数据空值一共有两种情况，即缺失值和空值。其中，缺失值是指值实际存在，但没有存入值所属字段中，如成年人都有身份证，若某个成年人的身份证号属性值为空，就属于缺失值；空值是指因实际并不存在而空缺的值，如动物没有身份证，因此它们的身份证号属性为空。

数据异常指的是用统计分析的方法识别出异常值。计算某个字段的平均值、众数、中位数以及最大值、最小值等，可根据这些统计的值和相关的规则发现数据中的异常。

（2）重复数据。

重复数据是指一个现实实体在数据集合中以多个不完全相同的记录表示。检测重复数

据的方法有很多，例如基本的字段匹配、递归的字段匹配、Smith-Waterman 算法以及基于编辑距离的字段匹配等方法。

1.2 数据清洗的定义

数据清洗技术是提高数据质量的有效方法。这项技术是一个较新的研究领域，对大数据集的清洗工作需要花费很长的时间。由于不同的应用领域对数据清洗有不同的解释，因此数据清洗直到现在都没有一个公认、统一的定义。数据清洗主要应用于 3 个领域，即数据仓库领域、数据挖掘领域以及数据质量管理领域。

在数据仓库领域中，当多个数据库合并时或多个数据源进行集成时，都需要进行数据清洗。例如，当同一个实体的记录在不同数据源中以不同的表示格式或错误表示的情况下，合并后的数据仓库中就会出现重复的记录，数据清洗的程序就需要识别出重复的记录并消除重复的记录，也就是所谓的数据合并或清除(Merge/Purge)问题。在数据仓库环境中，数据清洗主要包括数据的清洗和结构的转换两个过程。

在数据挖掘领域中，数据清洗是数据进行预处理过程的第一个步骤。在数据预处理应用中，数据清洗的主要任务是提高数据的可用性，即去除噪声、无关数据以及空值等，并考虑数据的动态变化。在字符分类问题中，通过使用机器学习的技术进行数据清洗，即使用特定算法检测数据库对缺失和错误的数据予以修改。

在数据质量管理领域中，数据质量管理是一个学术界和商业界都感兴趣的领域。数据质量管理主要用于解决信息系统中的数据质量及集成问题。在该领域中，数据清洗从数据质量的角度出发，把数据清洗过程和数据生命周期集成在一起，对数据的正确性进行检查并提高数据质量。

1.3 数据清洗的原理

数据清洗是利用相关技术将“脏”数据转换为满足质量要求的数据。下面通过一张图描述数据清洗的原理，具体如图 1-3 所示。

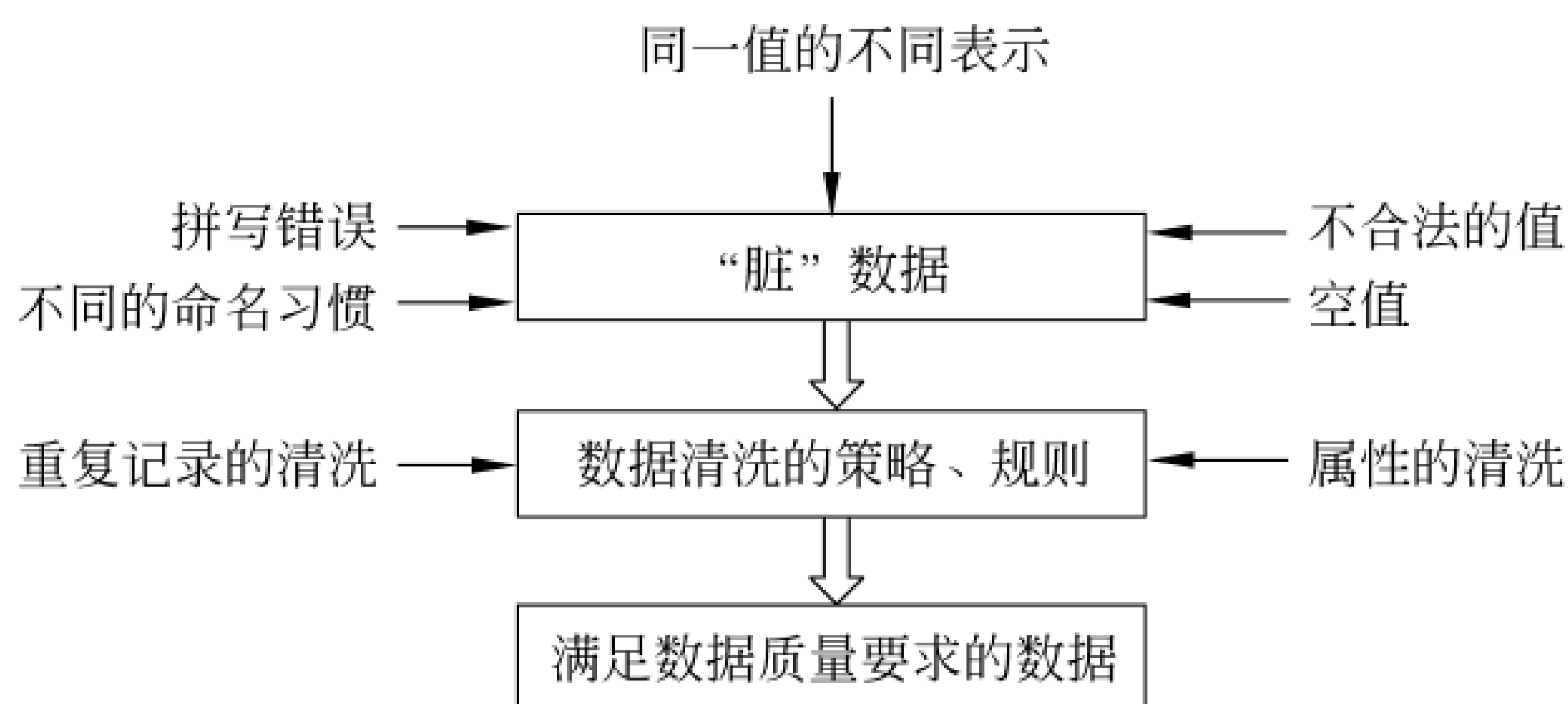


图 1-3 数据清洗的原理

从图 1-3 中可以看出，同一值的不同表示、拼写错误、不同的命名习惯、不合法的值以及空值都会导致“脏”数据出现，通过定义好的数据清洗策略和清洗规则(即数理统计技术、数

据挖掘技术等清洗策略)对“脏”数据进行清洗,得到满足数据质量要求的数据。

需要注意的是,数据清洗的目的是解决“脏”数据问题,即不是将“脏”数据洗掉,而是将“脏”数据洗干净。干净的数据指的是满足质量要求的数据。

1.4　数据清洗的基本流程

数据清洗的基本流程一共分为 5 个步骤,分别是数据分析、定义数据清洗的策略和规则、搜寻并确定错误实例、纠正发现的错误以及干净数据回流。下面通过一张图描述数据清洗的基本流程,具体如图 1-4 所示。

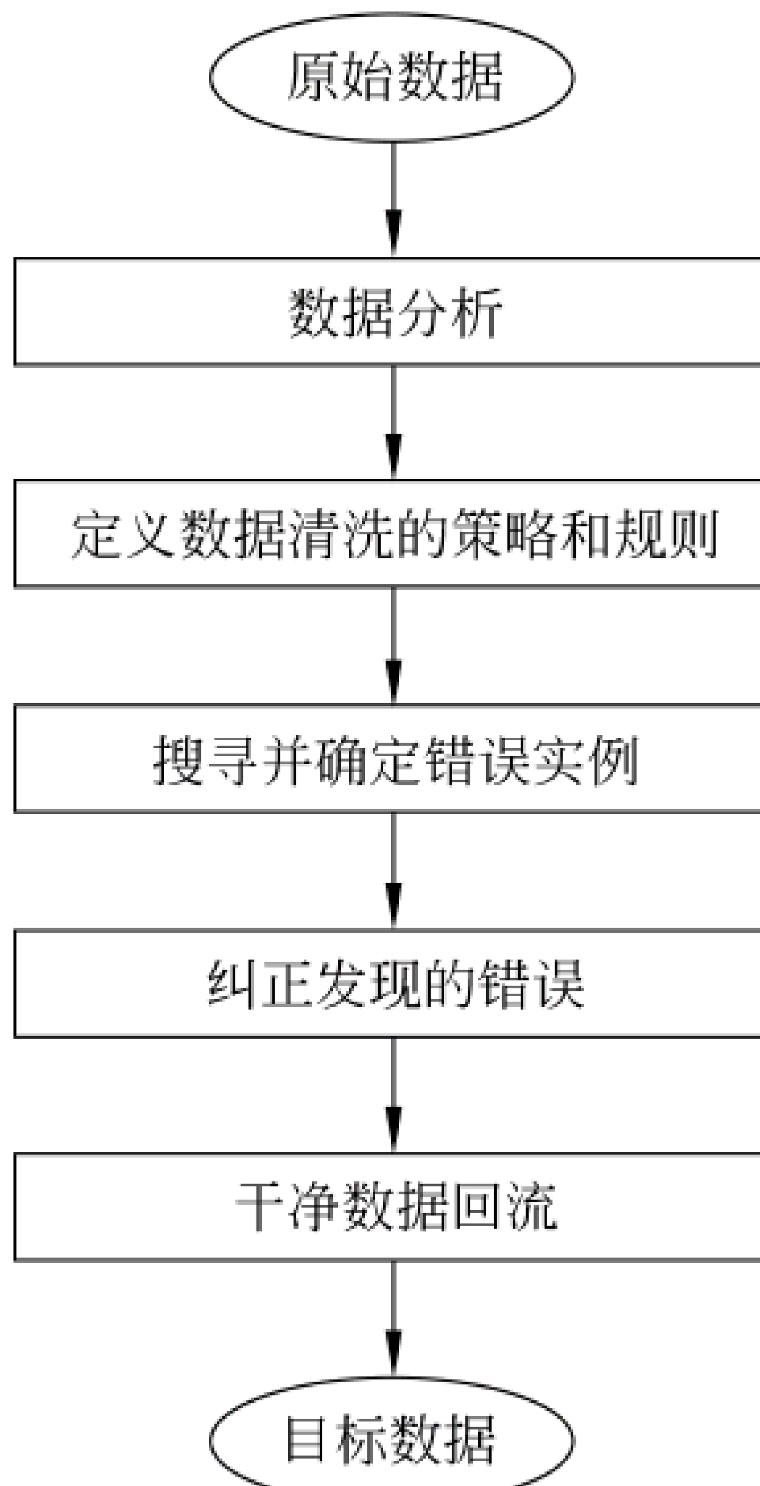


图 1-4　数据清洗的基本流程

接下来针对图 1-4 中数据清洗的基本流程进行详细讲解。

1. 数据分析

数据分析是数据清洗的前提和基础,通过人工检测或者计算机分析程序的方式对原始数据源的数据进行检测分析,从而得出原始数据源中存在的数据质量问题。

2. 定义数据清洗的策略和规则

根据数据分析出的数据源个数和数据源中的“脏”数据程度定义数据清洗策略和规则,并选择合适的数据清洗算法。

3. 搜寻并确定错误实例

搜寻并确定错误实例步骤包括自动检测属性错误和检测重复记录的算法。

手工检测数据集中的属性错误需要花费大量的时间、精力以及物力,并且该过程本身很容易出错,所以需要使用高效的方法自动检测数据集中的属性错误,主要检测方法有基于统计的方法、聚类方法和关联规则方法。

检测重复记录的算法可以对两个数据集或者一个合并后的数据集进行检测,从而确定同一个现实实体的重复记录,即匹配过程。检测重复记录的算法有基本的字段匹配算法、递归字段匹配算法等。

4. 纠正发现的错误

根据不同的“脏”数据存在形式的不同,执行相应的数据清洗和转换步骤解决原始数据源中存在的质量问题。需要注意的是,对原始数据源进行数据清洗时,应该将原始数据源进行备份,以防需要撤销清洗操作。

为了便于处理单数据源、多数据源以及单数据源与其他数据源合并的数据质量问题,一般需要在各个数据源上进行数据转换操作,具体如下。

(1) 从原始数据源的属性字段中抽取值(属性分离)。

原始数据源的属性一般包含很多信息,这些信息有时需要细化成多个属性,便于后续清

洗重复记录。

（2）确认并改正。

确认并改正输入和拼写的错误，然后尽可能地使该步骤自动化。若是基于字典查询拼写错误，则更利于发现拼写的错误。

（3）标准化。

为了便于记录实例匹配和合并，应该将属性值转换成统一格式。

5. 干净数据回流

当数据被清洗后，干净的数据替代原始数据源中的“脏”数据，这样可以提高信息系统的数据质量，还可避免将来再次抽取数据后进行重复的清洗工作。

1.5 数据清洗的策略

在数据仓库环境中，数据清洗可以在不同阶段实现，并且存在不同的清洗策略，目前数据清洗的策略主要分为一般的数据清洗策略和混合的数据清洗策略。下面针对数据清洗的两种策略进行讲解。

1. 一般的数据清洗策略

按照数据清洗的实现方式与范围划分，一般分为手工清洗策略、自动清洗策略、特定应用领域的清洗策略以及与特定应用领域无关的清洗策略，这4种清洗策略的具体介绍如下。

- 手工清洗策略，即通过人工直接修改“脏”数据。
- 自动清洗策略，即通过编写专门的应用程序检测并修改“脏”数据。
- 特定应用领域的清洗策略，即根据概率统计学原理检测并修改数值异常的记录。
- 与特定应用领域无关的清洗策略，即根据相关算法检测并删除重复记录。

2. 混合的数据清洗策略

混合的数据清洗策略主要以自动清洗为主。在数据仓库的数据初次装载阶段和增量装载阶段，可以通过编写应用程序实现批量数据的自动清洗，但该清洗策略并不能完全涵盖所有的错误类型。若无法按照已有策略识别某些错误类型，修改数据的工作就需要人工监督和确认，这时系统会设定异常报警功能，通过用户自身对错误的识别、理解和确认，最终实现数据清洗。

1.6 常见的数据清洗方法

常见的数据质量问题主要包括缺失值、重复值以及错误值等问题。下面针对缺失值的清洗、重复值的清洗以及错误值的清洗进行讲解。

1. 缺失值的清洗

缺失值的清洗方法主要分为两类，即忽略缺失值数据和填充缺失值数据。

(1) 忽略缺失值数据方法是直接通过删除属性或实例忽略缺失值的数据。

(2) 填充缺失值数据方法是使用最接近缺失值的值替代缺失的值,包括人工填写缺失值,使用一个全局常量填充空缺值(即将缺失的值用同一个常量 Unknown 替换)以及使用属性的平均值、中间值、最大(小)值填充缺失值,或使用最可能的值(即通过回归、贝叶斯形式化方法的工具或决策树归纳确定的值)填充缺失值。

2. 重复值的清洗

目前清洗重复值的基本思想是“排序和合并”。清洗重复值的方法主要有相似度计算和基于基本近邻排序算法等方法。

(1) 相似度计算是通过计算记录的个别属性的相似度,然后考虑每个属性的不同权重值,进行加权平均后得到记录的相似度,若两个记录相似度超过某一个阈值,则认为两条记录匹配,否则认为这两条记录指向不同的实体。

(2) 基于基本近邻排序算法的核心思想是为了减少记录的比较次数,在按关键字排序后的数据集上移动一个大小固定的窗口,通过检测窗口内的记录判定它们是否相似,从而确定并处理重复记录。

3. 错误值的清洗

错误值的清洗方法主要包括使用统计分析的方法识别可能的错误值(如偏差分析、识别不遵守分布或回归方程的值)、使用简单规则库(即常识性规则、业务特定规则等)检测出错误值、使用不同属性间的约束以及使用外部的数据等方法检测和处理错误值。

1.7　本章小结

本章主要讲解了数据预处理的相关知识,包括数据质量概述、数据质量的评价指标、数据质量的问题分类以及数据清洗的定义、数据清洗的原理、数据清洗的基本流程、数据清洗的策略和常见的数据清洗方法,希望读者通过本章的学习,可以对数据预处理有基本的认识,便于后续章节的学习。

1.8　本章习题

一、填空题

1. 对原始数据进行有效的________是大数据分析和应用过程中的关键环节。
2. 数据质量的评价指标有准确性 ________、简洁性、________。
3. 数据质量的问题可以分为两类,分别是________和基于清洗方式的“脏”数据分类。
4. ________技术是提高数据质量的有效方法。
5. 常见的数据质量问题主要包括缺失值、________以及错误值等问题。

二、判断题

1. 直接使用原始数据不会影响数据决策的准确性和效率。　(　　)

2. 从数据清洗方式的设计者角度看，可以将“脏”数据分为“独立型‘脏’数据”和“依赖型‘脏’数据”两类。 ( )

3. 依赖型“脏”数据主要包括缺失数据和拼写错误数据等“脏”数据。 ( )

4. 数据清洗的目的是要将“脏”数据洗掉。 ( )

5. 基于数据源的“脏”数据分类的数据质量问题可以分为单数据源问题和多数据源问题。 ( )

三、选择题

1. 下列选项中，________是评价数据质量的核心准则。

   A. 完整性　　B. 准确性　　C. 适用性　　D. 简洁性

2. 下列策略中，________策略属于一般的数据清洗策略。

   A. 手工清洗　　B. 自动清洗

   C. 特定应用领域　　D. 与特定应用领域无关

3. 下列说法中，关于清洗重复值的说法正确的是________。

   A. 清洗重复值的基本思想是“分而合之”

   B. 清洗重复值的基本思想是“排序”

   C. 清洗重复值的基本思想是“排序和合并”

   D. 清洗重复值的基本思想是“合并”

四、简答题

简述数据清洗的基本流程。

第 2 章 初识ETL

学习目标

(1) 了解 ETL 的概念
(2) 理解 ETL 的体系结构
(3) 熟悉 ETL 的关键技术
(4) 掌握 ETL 的常见工具

对于企业来说,数据已经成为一种重要的战略资源,为了充分利用好自己的数据资源,使用 ETL 技术进行数据分析已成为企业决策的重要工作内容之一。ETL 是将业务系统的数据经过抽取、清洗转换之后加载到数据仓库的过程,目的是将企业中的不完整数据、重复数据以及错误数据等“脏”数据内容通过清洗转换操作转变为符合企业要求的数据,便于为企业的决策提供分析依据。本章将针对 ETL 的相关知识进行详细讲解。

2.1 基于 ETL 的数据清洗

企业每年产生海量的数据,如何从海量数据中挖掘有价值的数据成为大数据研究的一个重点。基于 ETL 的数据清洗是挖掘有价值数据的一种方案,接下来本节将针对 ETL 的概念、体系结构和设计进行讲解。

2.1.1 ETL 的概念

ETL 是英文 Extract-Transform-Load 的缩写,用来描述将数据从源端经过抽取(extract)、转换(transform)、加载(load)至目的端的过程,它能够对各种分布的、异构的源数据(如关系数据)进行抽取,按照预先设计的规则将不完整数据、重复数据以及错误数据等“脏”数据内容进行清洗,得到符合要求的“干净”数据,并加载到数据仓库中进行存储,这些“干净”数据就成为了数据分析、数据挖掘的基石。

ETL 是实现商务智能(Business Intelligence,BI)的核心。一般情况下,ETL 会花费整个 BI 项目三分之一的时间,因此 ETL 设计得好坏直接影响 BI 项目的成败。

企业中常用的 ETL 实现有多种方式,常见的方式如下。

(1) 借助 ETL 工具(如 Pentaho Kettle、Informatic 等)。
(2) 编写 SQL 语句。
(3) 将 ETL 工具和 SQL 语句结合起来使用。

上述 3 种实现方式各有利弊,其中第 1 种方式可以快速建立 ETL 工程,屏蔽复杂的编码任务、加快速度和降低难度,但是缺少灵活性;第 2 种方式使用编写 SQL 语句的方式优点是灵活,可以提高 ETL 的运行效率,但是编码复杂,对技术要求比较高;第 3 种方式综合了前面两种方法的优点,可以极大地提高 ETL 的开发速度和效率。

2.1.2 ETL 的体系结构

ETL 主要是用来实现异构数据源数据集成的。多种数据源的所有原始数据大部分未作修改就被载入 ETL,因而,无论数据源在关系型数据库、非关系型数据库,还是在外部文件,集成后的数据都将被置于数据库的数据表或数据仓库的维度表中,以便在数据库内或数据仓库中作进一步转换(因此,一般会将最终的数据存储到数据库或者数据仓库中)。ETL 的体系结构如图 2-1 所示。

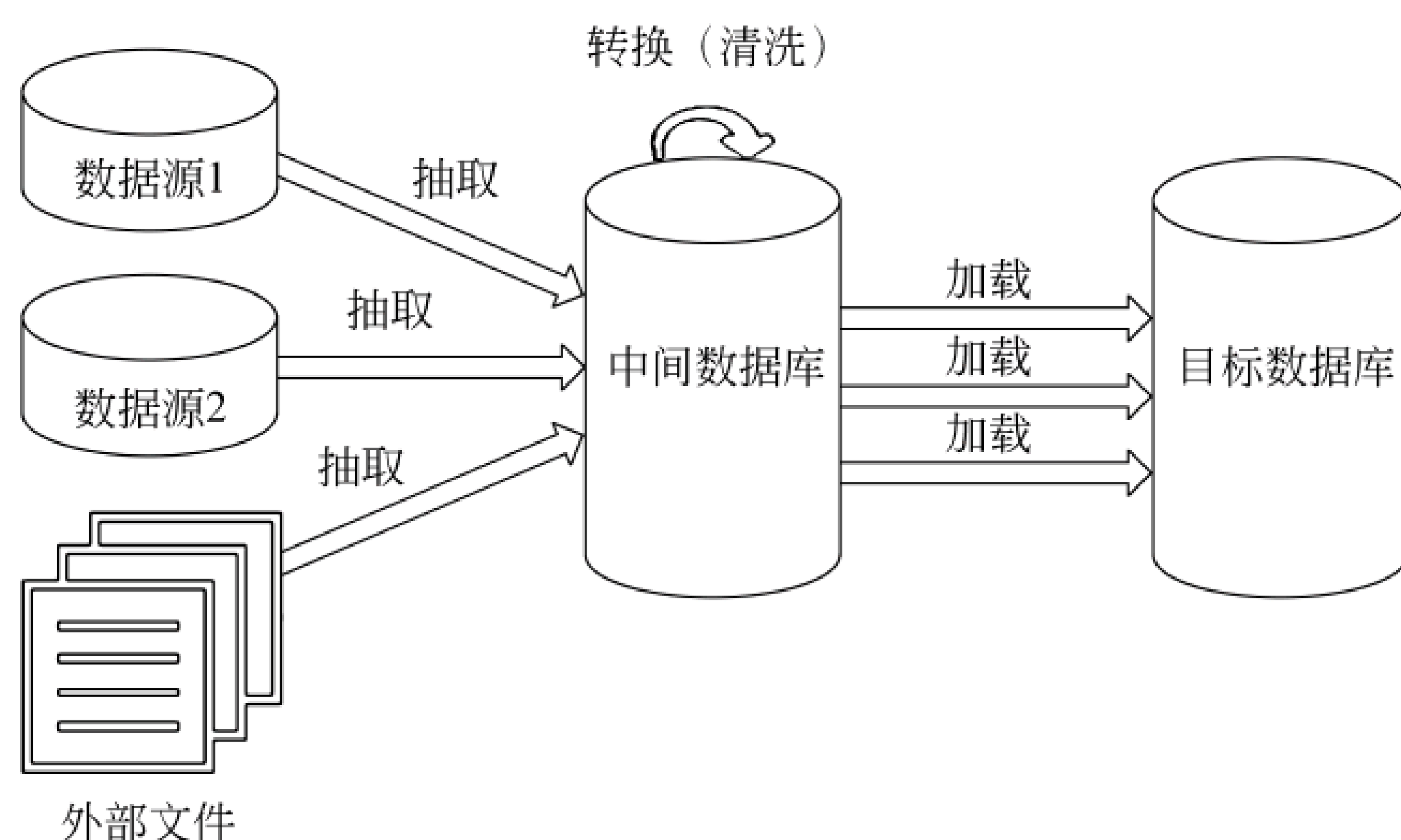


图 2-1 ETL 的体系结构

在图 2-1 中,若数据源 1 和数据源 2 均为功能较强大的 DBMS(数据库管理系统),则可以使用 SQL 语句完成一部分数据清洗工作。但是,如果数据源为外部文件,就无法使用 SQL 语句进行数据清洗工作了,只能直接从数据源中抽取出来,然后在数据转换的时候进行数据清洗的工作。因此,数据仓库中的数据清洗工作主要还是在数据转换的时候进行。清洗好的数据将保存到目标数据库中,用于后续的数据分析、数据挖掘以及商业智能。

2.2 ETL 关键技术

ETL 关键技术一共有 3 个,分别是数据的抽取、数据的清洗转换以及数据的加载。本节针对数据的抽取、数据的清洗转换、数据的加载进行详细介绍。

2.2.1 数据的抽取

数据的抽取就是从异构数据源抽取数据,但是并不是所有数据源中的数据都有实际的价值。业务人员和设计人员需要分析讨论哪些数据有价值,哪些数据可以忽略,然后制定抽取策略。数据的抽取分为数据的全量抽取和数据的增量抽取。其中,全量抽取类似于数据迁移或数据复制,它将原数据表中的数据全部抽取出来;经过上次抽取后,源数据表中的数

据出现变化时，会进行增量抽取。增量抽取是抽取数据源表中新增或被修改的数据。

在 ETL 的使用过程中，数据的增量抽取比数据的全量抽取应用更广泛。要实现增量抽取，就要准确地捕获到数据库中数据源表数据的变化，因此捕获变化的数据是增量抽取的关键。数据的增量抽取有 4 种方式，具体如下。

1. 触发器方式

触发器方式是根据抽取要求，在要被抽取的数据源表上建立插入、修改、删除 3 个触发器，每当数据源表中的数据发生变化，就被相应的触发器将变化的数据写入一个增量日志表中。ETL 的增量抽取则是从增量日志中抽取，而不是直接在源表中抽取数据，同时，增量日志表中抽取过的数据要及时被标记或者删除。

2. 时间戳方式

时间戳方式是指增量抽取时，抽取进程通过比较指定抽取时间与抽取源表的时间戳字段的值决定抽取哪些数据。这种方式需要在源表中增加一个时间戳字段，系统中更新或修改源表数据的时候，也会同时修改时间戳字段的值。插入数据的时间戳由系统时间指定。

3. 全表比对方式

全表比对方式是指在增量抽取时，ETL 进程逐条比较源表和目标表的记录，将新增或修改等变化的记录过滤读取出来。

4. 日志表方式

对于建立了业务系统的生产数据库的企业来说，可以在数据库中创建业务（企业中的业务）日志表，当特定需要监控的业务数据发生变化时，由相应的业务系统程序模块更新维护日志表的内容。增量抽取时，通过读日志表数据决定加载哪些数据及如何加载。日志表的维护需要由业务系统程序编写代码完成。

以上 4 种常见的增量抽取方式没有一种方式具有绝对的优势，不同的方式在不同企业中的表现大体都是相对平衡的。通常根据企业中的业务需求和硬件环境选择 ETL 抽取机制。

2.2.2　数据的清洗转换

数据的清洗转换是指将抽取到的数据源表中的数据，根据数据仓库系统模型的要求进行数据的清洗、转换等操作，保证来自不同系统、不同格式数据的一致性和完整性，并且要按照业务要求加载到目标表。数据的清洗转换是 ETL 中最复杂的部分，主要任务是过滤掉不符合要求的数据。不符合要求的数据主要是有不完整的数据、错误的数据、重复的数据三大类。下面针对不符合要求的三大类数据进行详细介绍。

1. 不完整的数据

数据上报、接口调用时都会产生大量的不完整数据，不完整数据的产生是不可避免的现

象,而不完整的数据对大数据环境下的决策具有一定的影响。不完整数据主要包括缺失部分信息的数据。检测不完整数据的方法具体如下。

缺失部分或全部内容的数据主要是采用计算机和人工相结合的方法进行查找,并对缺失的内容进行填充处理。不完整数据的清洗流程如图 2-2 所示。

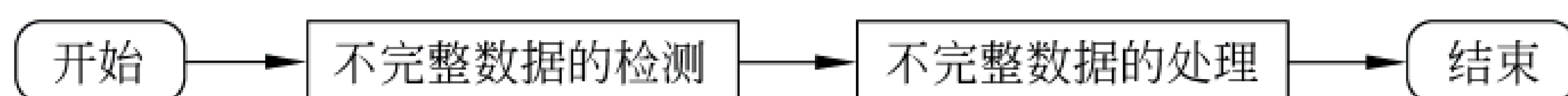


图 2-2 不完整数据的清洗流程

在图 2-2 中,不完整数据的清洗流程主要分为 3 个步骤,具体如下。

(1) 对获得的数据源进行不完整数据的检测,为后续的数据处理提供所需的数据。

(2) 对检测出来的不完整数据进行处理,如修复缺失部分或全部内容的数据。

(3) 输出处理后的符合要求的完整数据。

2. 错误的数据

大数据环境下数据量的剧增使得获取到的数据源会由于各种原因存在大量的错误数据。

错误数据产生的原因是业务系统不够健全,在接收输入数据后没有进行过滤判断,而是直接将数据写入后台数据库造成的,如数值数据输成全角数字字符、字符串数据后面出现一个回车操作、日期格式不正确、日期越界等错误。错误数据的清洗流程如图 2-3 所示。

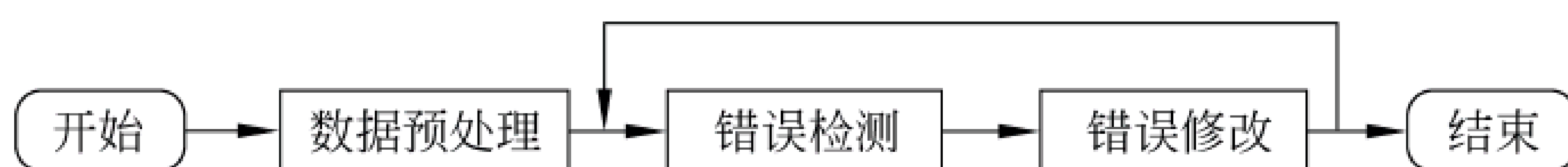


图 2-3 错误数据的清洗流程

在图 2-3 中,错误数据的清洗流程主要分为 3 个步骤,具体如下。

(1) 将数据源按照规定的数据格式进行检测,并执行数据预处理,为后续的处理步骤做准备。

(2) 对预处理后的数据进行一致性检测,如果预处理后的数据与原始数据存在完整性不一致的问题,则通过数据修改过程使数据统一。为避免再次出现该问题,应重复进行检测与修改过程,直到符合要求为止。

(3) 输出修改后的数据。

3. 重复的数据

产生重复的数据原因较多,如数据集成、系统重复录入等,通常表现为多条记录表达的含义相同,或同一目标实体的记录虽然在形式上有所不同,但其描述的目标却相同。这些重复记录的数据特征并不明显,但是对数据识别和数据清洗造成了很大的难度。因此,对重复记录数据进行清洗,可以提高数据库的使用率,降低系统消耗,并提高数据的质量。

重复数据检测主要分为基于字段和基于记录的重复检测。基于字段的重复检测算法主要为编辑距离算法;基于记录的重复检测算法主要包括排序邻居算法、优先队列算法、N-

Gram 聚类算法。采用排序合并算法清洗重复数据的流程如图 2-4 所示。

在图 2-4 中，重复数据的清洗流程主要分为 4 个步骤，具体如下。

（1）通过对源数据库属性段的分析，找到属性的唯一值，并根据唯一值对源数据库中的数据记录进行排序，可以选择自上而下或者自下而上的顺序排序。

（2）按顺序扫描数据库中的每一条记录，并将它与相邻的记录进行比较，进行记录的相似度匹配计算，输出修改后的数据。

（3）如果计算出的相似度数值大于系统设定的阈值，说明该记录或连续的几条记录为相似重复记录，则进行数据记录的合并或删除操作；否则扫描下一条数据记录，重复以上第（2）、（3）步骤。

（4）当所有数据记录检测完毕后，输出清洗后的数据结果。

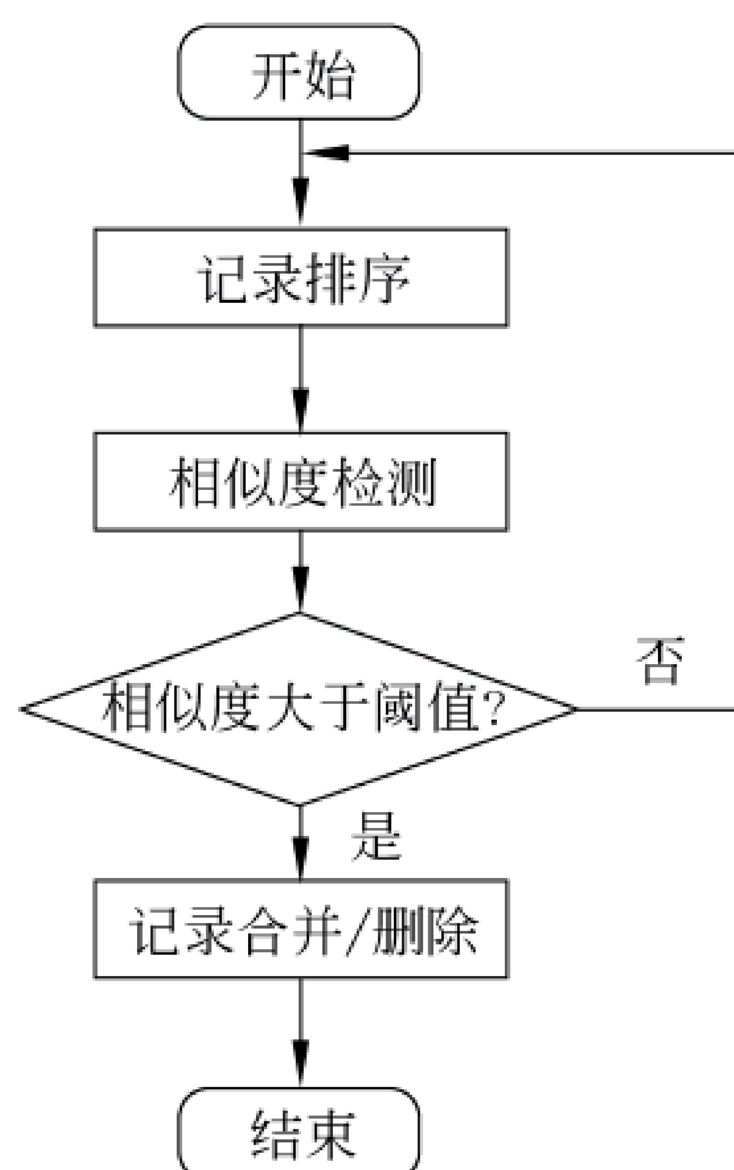


图 2-4　采用排序合并算法清洗重复数据的流程

2.2.3　数据的加载

数据的加载是 ETL 的最后一个阶段，它的主要任务是将数据从临时数据表或文件中加载到指定的数据仓库中。一般来说，可以通过编写 SQL 语句和利用加载工具将数据加载到数据仓库中。ETL 的数据加载和数据抽取类似，将数据加载到目标数据表或者数据仓库的过程中可分为全量加载、增量加载以及批量加载。下面详细介绍一下 ETL 的全量加载、增量加载以及批量加载。

1. 全量加载

全量加载是指全表删除后再进行全部（全量）数据加载。从技术角度来说，全量加载和增量加载相比，全量加载更简单。一般只需要在数据加载之前将目标表清空，再将源数据表中的数据进行导入。但是，由于数据量、系统资源和数据实时性的要求，很多情况下都需要使用增量加载机制。

2. 增量加载

增量加载是指目标表仅更新源数据表中变化的数据。增量加载的关键在于如何正确设计相应的方法，用于从源数据表中抽取增量的数据，以及变化“牵连”数据（虽没有变化，但受到变化数据影响的数据）。同时，将这些变化的和未变化但受到影响的数据，在完成相应的逻辑转换后更新到数据仓库中。

数据加载的性能和作业失败后可进行恢复重启的易维护性，需要一个有效的增量抽取机制的支持。因为在一个有效的增量抽取机制，ETL 能够将业务系统中变化的数据按一定的频率准确地进行捕获，并且不会对业务系统造成太大的压力，也不会影响现有的业务。

ETL 的增量加载类似于增量抽取，同样有 4 种方式，分别是时间戳方式、日志表方式、全表对比方式、全表删除插入方式，具体介绍如下所示。

(1) 时间戳方式，即在业务表中统一添加一个字段作为时间戳，当联机分析处理(OLAP)系统更新修改业务数据时，同时也会修改时间戳字段值，这时就将更新修改的数据加载到目标表中。

(2) 日志表方式，即在OLAP系统中添加日志表，业务数据发生变化时，更新维护日志表内容。

(3) 全表对比方式，即抽取所有源数据，在加载目标表之前先根据主键和字段进行数据比对，有更新的数据就进行更新或插入。

(4) 全表删除插入方式，删除目标表中的数据，将源数据表中的数据全部加载到目标表中。

3. 批量加载

通常，对于几十万条记录的数据迁移而言，采取DML(数据操纵语言)的insert、update、delete等语句能够较好地将数据迁移到目标数据库中。然而，当数据迁移量过大时，DML语句执行时生成的事物日志(事务日志是一个与数据库文件分开的文件，用于存储对数据库进行的所有更改，并全部记录插入、更新、删除、提交、回退和数据库模式变化)和约束条件将大大影响加载性能，故需要针对数据采取批量加载处理。

2.3 ETL常见工具介绍

目前比较流行的ETL工具有Pentaho Kettle、Hawk、Informatica PowerCenter及DataStage，对这些工具的介绍如下。

1. Pentaho Kettle

Pentaho Kettle是一款国外免费开源的ETL工具，纯Java语言编写，可以在Windows、Linux、UNIX系统上运行，并且是绿色无需安装的。

Kettle的中文名称叫水壶，该工具的设计理念是希望把来自不同数据库中的数据放到一个“壶”里，然后以一种指定的格式流出。Kettle拥有两种脚本文件，分别是Transformation(转换)和Job(作业)，其中Transformation是用于完成数据的基础转换，而Job是完成整个工作流的控制。

2. Hawk

Hawk是一种数据采集和清洗工具，依据GPL(GNU通用公共许可证)协议开源，基于C#语言编写的，并且其前端界面使用WPF开发，支持插件扩展。

Hawk的含义为“鹰”，能够高效、准确地捕杀猎物。也就是说，Hawk能够灵活、有效地采集来自网页、数据库和文件等来源的数据，并通过可视化的拖曳操作快速地进行生成、过滤及转换等操作。Hawk主要应用于爬虫和数据清洗等领域。

3. Informatica PowerCenter

Informatica PowerCenter是Informatica公司开发的世界级的企业数据集成平台，也

是业界领先的 ETL 工具。Informatica PowerCenter 用于访问和集成几乎任何业务系统、任何格式的数据，它可按任意速度在企业内交付数据，具有高性能、高可扩展性、高可用性的特点。Informatica PowerCenter 提供了多个可选的组件，以扩展 Informatica PowerCenter 的核心数据集成功能，这些组件包括数据清洗和匹配、数据屏蔽、数据验证、元数据交换等。

4. DataStage

IBM 的 InfoSphere DataStage 简称 DataStage，它是一个领先的 ETL 平台，可跨多个企业系统集成数据。DataStage 利用高性能并行框架，可根据项目需求在云中或者本地部署 ETL 环境，它支持 HBase、Hive、Amazon 以及 MongoDB 等数据库的连接，可以灵活、有效地更新和管理数据继承的基础架构。

2.4　本章小结

本章主要讲解了 ETL 的相关知识，包括基于 ETL 的数据清洗、ETL 关键技术以及 ETL 常见的工具。希望读者通过本章的学习认识 ETL，并熟悉 ETL 的关键技术和掌握 ETL 常见的工具，便于后续章节的学习。

2.5　本章习题

一、填空题

1. ________是实现商务智能(Business Intelligence，BI)的核心和灵魂。
2. ETL 是将业务系统的数据经过抽取、________之后加载到数据仓库的过程。
3. ETL 的实现有多种方式，常见的方式有借助________、编写 SQL 语句、________。
4. 数据的抽取分为数据的全量抽取和数据的________。
5. 不符合要求的数据主要有不完整的数据、________、重复的数据三大类。

二、判断题

1. 基于 ETL 的数据清洗是挖掘有价值数据的一种方案。　(　　)
2. 如果数据源为外部文件，可使用 SQL 语句进行数据清洗工作。　(　　)
3. 不完整数据主要包括日期越界的数据。　(　　)
4. 重复数据检测主要分为基于字段和基于记录的重复检测。　(　　)
5. Kettle 是一款国外免费开源的 ETL 工具，纯 Python 语言编写。　(　　)

三、选择题

1. 下列方式，________不属于增量抽取的方式。
   A. 触发器方式　　B. 时间戳方式
   C. 全表比对方式　　D. 批量抽取方式

2．下列算法中，________不可用于检测重复记录。

A．编辑距离算法　　　　B．优先队列算法

C．N-Gram 聚类算法　　　　D．排序邻居算法

四、简答题

简述不符合要求数据的清洗流程。

第3章 Kettle工具的基本使用

学习目标

(1) 了解 Kettle 工具
(2) 掌握 Kettle 的下载安装
(3) 熟悉 Kettle 的基本概念
(4) 掌握 Kettle 的基本功能

“工欲善其事,必先利其器”,Kettle 作为一款开源的 ETL 解决方案,掌握它的基本用法非常有必要。本章将针对 Kettle 工具的相关知识进行详细讲解。

3.1 Kettle 简介

3.1.1 Kettle 概述

Kettle 是一款国外免费开源的轻量级 ETL 工具,是基于 Java 语言开发的,可以在 Windows、Linux、UNIX 系统上运行,并且是绿色无需安装的,可用于各种数据库之间数据的迁移。

Kettle 的中文名称为“水壶”,其设计理念是主程序员 Matt 希望将来自不同数据库中的数据放到一个壶里,然后以一种指定的格式流出(即按照用户要求的格式输出)。Kettle 支持管理来自不同数据库的数据,通过提供一个图形化的用户环境描述用户想要做什么,而不是用户想要怎么做。

Kettle 工具主要由 4 个组件组成,分别是 Spoon、Pan、Kitchen 及 Carte 组件,具体功能介绍如下。

- Spoon 是 Kettle 的集成开发环境,它会提供一个基于 SWT 的图形用户界面,主要用于构建 ETL Jobs(作业)和 Transformations(转换),也可用于执行或调试作业、转换,还可用于监控 ETL 操作的性能。
- Pan 是以命令行的方式(即编写 Shell 脚本)执行 Spoon 生成的 Transformations 程序,运行在后台,并且该组件没有图形化用户界面。
- Kitchen 是以命令行的方式(即编写 Shell 脚本)执行 Spoon 生成的 Jobs 程序,运行在后台,并且该组件没有图形化用户界面。
- Carte 是 Kettle 中的一个重要组件,它是基于 Jetty 的轻量级 HTTP 服务器,运行在后台,主要用于远程监控 HTTP 执行 Jobs 和 Transformations 的进度。

3.1.2 Kettle 的设计原则

每个 ETL 工具都会有自己的设计原则,Kettle 也不例外。Kettle 的设计原则一共有 7 点,具体内容如下。

1. 易于开发

作为数据仓库和 ETL 的开发者,如果只想把时间用在创建 BI 解决方案上,那么任何用于软件安装和配置的时间都是一种浪费。例如,为了创建数据库连接,很多与 Kettle 类似的工具都要求用户手工输入数据库驱动的类名和 JDBC URL 连接串,虽然用户可以通过互联网搜索到这些信息,但这明显把用户的注意力转移到了技术方面,并非业务方面,而 Kettle 就是尽量避免这类问题出现。

2. 避免自定义开发

一般来说,ETL 工具的作用是使复杂的事情变得简单,简单的事情更简单。ETL 提供了标准化的构建组件满足 ETL 开发人员不断重复的需求,通过手工编写 Java 代码或 Java 脚本代码实现一些功能,但是增加的代码会给项目增加复杂度和维护成本,因此要尽量避免手工开发,可组合使用已提供的组件完成任务。

3. 所有功能都能通过用户界面完成

对于"所有功能都能通过用户界面完成"这一黄金准则也有几个例外(如 kettle. properties 和 shared. xmr 文件就是两个例外,不能通过 Kettle 界面修改这两个配置文件,而是需要通过手工修改),如果不直接把所有功能通过界面的方式提供给用户,那么就是在浪费开发人员的时间,也是在浪费用户的时间。

4. 没有命名限制

ETL 转换里有各种各样的名称,如数据库连接、转换、步骤、数据字段、作业等都有一个名称。若在命名时考虑到一些限制(如长度、选择的字符),就会使工作变得烦琐。ETL 只需要足够智能化的处理 ETL 开发人员设置的各种名称。

5. 透明

如果有 ETL 工具需要了解转换中某一部分工作是如何完成的,那么这个 ETL 工具就是不透明的。若想实现 ETL 工具里的某一个功能,就需要准确地知道这个功能是如何完成的。允许用户看到 ETL 过程中各部分的运行状态也很重要,这样可以加快开发速度,降低维护成本。

6. 灵活的数据通道

对 ETL 开发者来说,创造性极为重要,不但可以让你享受到工作的乐趣,而且还能让你以最快的方式开发出 ETL 方案。Kettle 在数据的发送、接收方式上设计得尽可能灵活。Kettle 可以在文本文件、关系数据库等不同数据源之间复制和分发数据。

7. 只映射需要映射的字段

在一些 ETL 工具里可以看到数百行的输入和输出映射，对于维护人员来说，这是一个很强大的功能。在 ETL 开发过程中，字段在不断地变化，大量的字段映射也会增加维护的成本，而 Kettle 的一个核心原则是将 ETL 流程中所有未指定的字段自动传递到下一个组件中，因此极大地降低了维护的成本。也就是说，输入的字段会自动出现在输出流中，除非中间过程专门设置了终止某个字段的传递。

3.2　Kettle 的下载安装

Kettle 的集成开发环境 Spoon 提供了一个基于 SWT 的图形用户界面，主要用于 ETL 的开发。下面分步骤讲解如何下载安装 Windows 环境下的 Kettle 工具。由于 Kettle 工具是运行在 JVM 平台上的，所以安装 Kettle 之前必须配置好 JDK 环境。关于 JDK 环境的下载、安装以及配置，这里不再赘述（需要注意的是 Kettle 版本和 JDK 版本的兼容性）。Kettle 的下载安装步骤具体如下。

1. 下载 Kettle 安装包

Kettle 官网下载地址为 https://sourceforge.net/projects/pentaho/files/Data%20Integration/。由于编写本书时，Kettle 工具的最新版本是 pdi-ce-8.2.0.0-342.zip，所以本书就以 Kettle 8.2.0 为准进行下载安装，具体如图 3-1 所示。

Summary　Files　Reviews　Support　Wiki　News　Donate

Download Latest Version pdi-ce-8.2.0.0-342.zip (1.2 GB)　Get Updates

Home / Data Integration

Name	Modified	Size	Downloads / Week
Parent folder			
7.1	2017-05-22		744
7.0	2016-11-09		132
6.1	2016-04-13		124
6.0	2015-12-10		56
5.4	2015-06-15		57
5.3	2015-02-17		13
5.2	2014-10-06		17
5.1	2014-06-24		18

图 3-1　Kettle 版本的选择

单击图 3-1 中的 Get Updates 按钮，下载 Kettle 工具。

2. 安装 Kettle

由于 Kettle 工具是绿色无需安装的，因此只要解压下载 Kettle 工具 pdi-ce-8.2.0.0-

342.zip 即可，解压后 Kettle 的安装目录为/pdi-ce-8.2.0.0-342/data-integration，具体如图 3-2 所示。



图 3-2 Kettle 的安装目录

3. 配置 Kettle

将 Java 和 Kettle 的安装路径都添加至系统环境变量中，便于后续在 Windows 任何位置都可进行引用启动 Kettle 工具；将数据库驱动（本书使用 mysql-connector-java-5.1.46-bin.jar 驱动）添加至 Kettle 安装包下的 lib 文件夹下，避免创建数据库连接时出现数据库找不到的问题。

4. 启动 Kettle

双击 Kettle 安装目录下的 Spoon.bat 脚本，启动 Kettle。通过查看 Kettle 启动的界面，判断 Kettle 工具是否启动成功，若出现图 3-3 所示的界面，则说明 Kettle 安装启动成功。

Kettle 的主界面大致分为 4 部分，即工具栏、工具图标、Kettle 的树形列表以及工作区，具体如图 3-4 所示。

在图 3-4 中，第一行红框是工具栏，主要有“文件”“编辑”“视图”“执行”“工具”以及“帮助”6 个操作选项；第二行红框是工具图标；左侧红框是 Kettle 的树形列表，主要包含“主对象树”和“核心对象”，其中“主对象树”包含转换和作业，而“核心对象”包含转换和作业各自对应的核心对象，且“核心对象”就是后续操作中使用到的步骤或控件；右侧红框是工作区，图 3-4 中显示的是一个欢迎界面，工作时关闭欢迎界面即可。

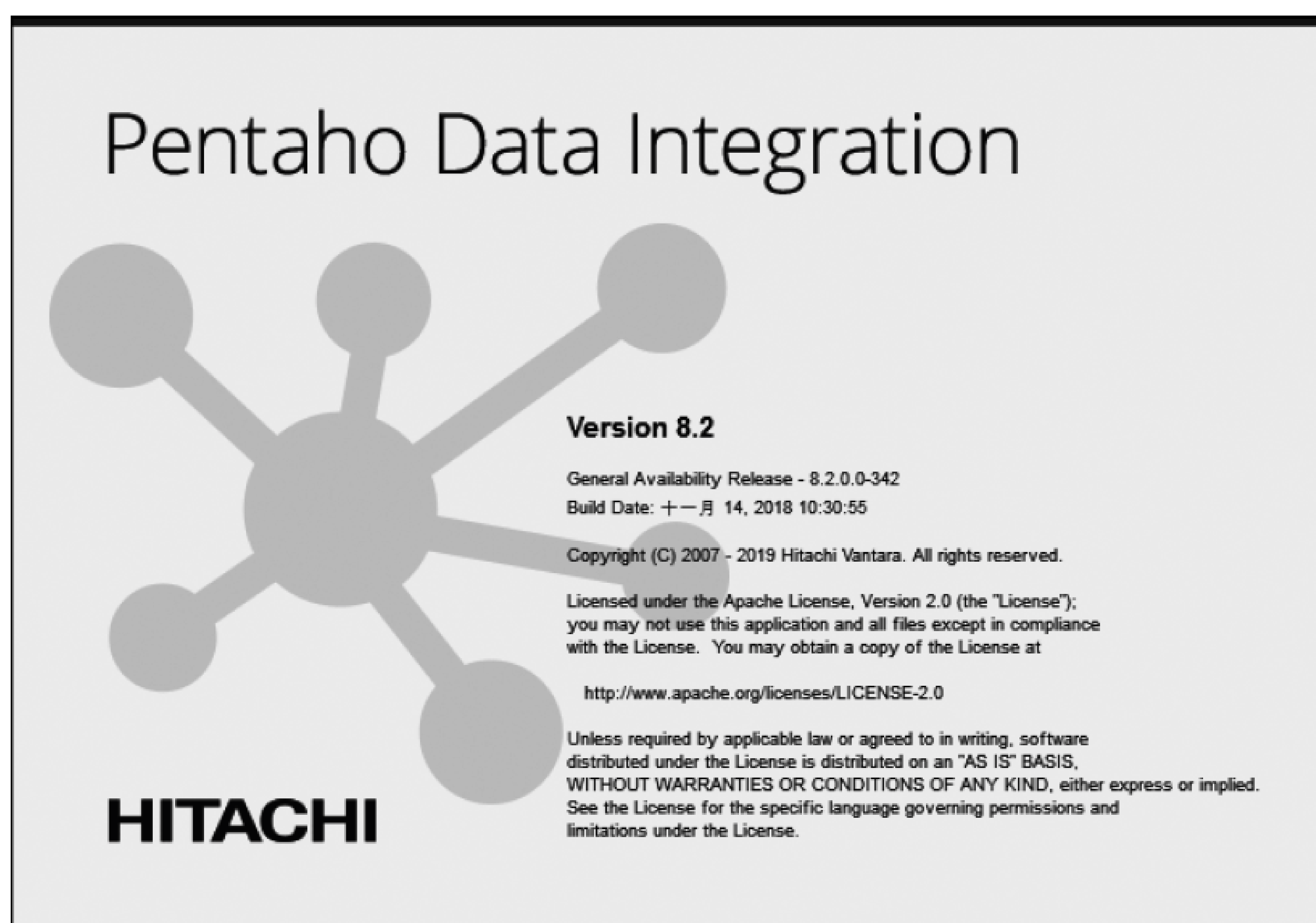


图 3-3　Kettle 安装启动成功的界面

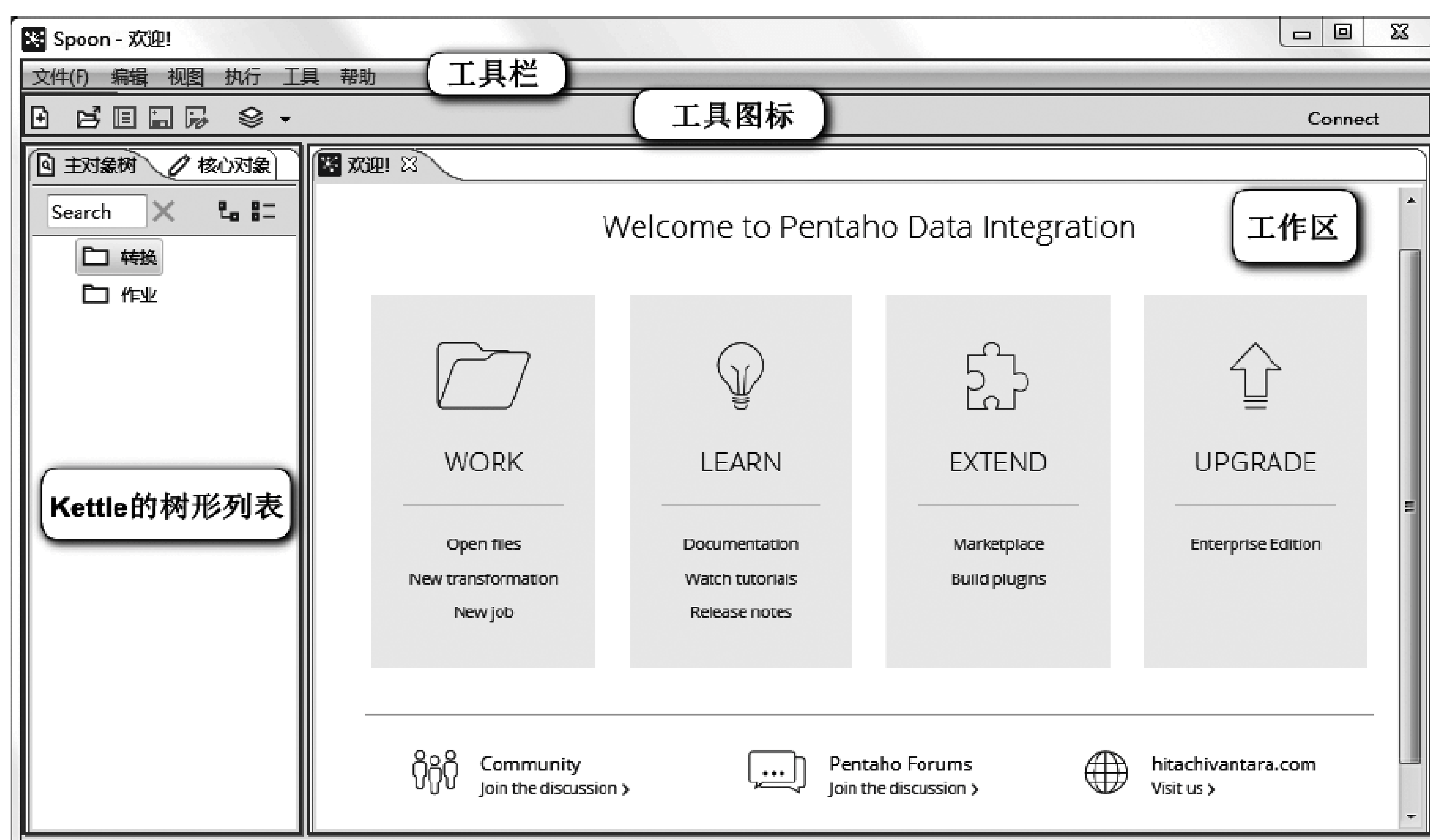


图 3-4　Kettle 的主界面

3.3　Kettle 的基本概念

一个数据抽取过程主要包括创建一个作业，并且每个作业可以包括多个转换操作。此数据抽取过程可通过 Kettle 工具完成，也可以通过编写程序调用的方式实现。下面通过一张图描述 Kettle 的概念模型，具体如图 3-5 所示。

从图 3-5 中可以看出，Kettle 工具的执行分为两个层次，即转换和作业，这两个层次最主要的区别在于数据传递和执行方式。接下来，对 Kettle 的转换、作业进行详细讲解。

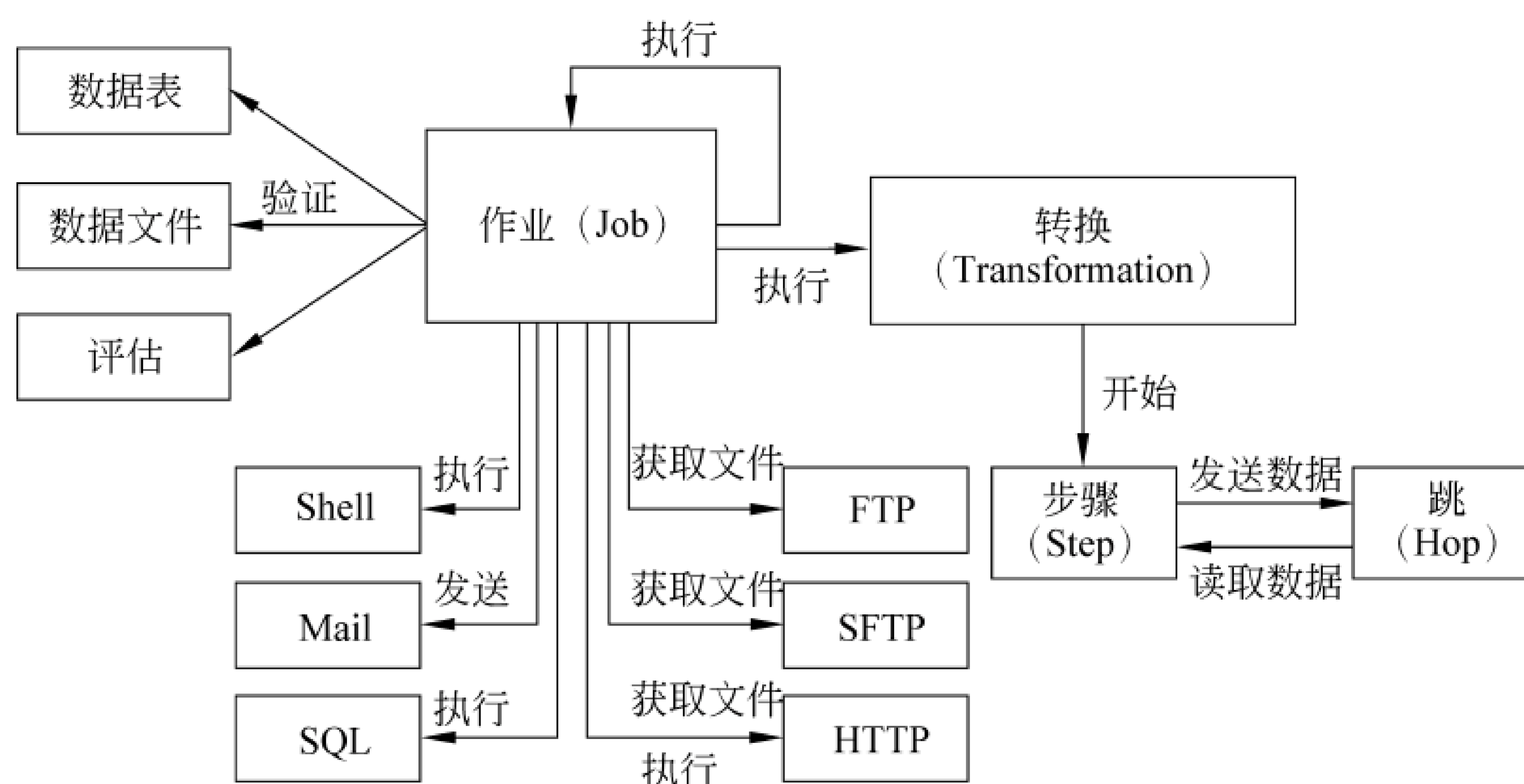


图 3-5 Kettle 的概念模型

3.3.1 转换

转换是 ETL 解决方案中重要的组成部分之一，主要用于数据的抽取、转换以及加载等操作，其本质是一组图形化的数据转换配置的逻辑结构。一个转换包括一个或多个步骤，如读取文件、过滤输出行、数据清洗或将数据加载到数据库中等步骤。转换中的步骤是通过跳连接的。跳定义了一个单向通道，允许数据从一个步骤向另一个步骤流动。在 Kettle 中，数据的单位是行，数据流就是数据行从一个步骤到另一个步骤的移动。

下面通过一个简单的例子详细讲解 Kettle 中的转换。

双击 Kettle 目录下的 Spoon. bat 脚本，启动 Kettle 工具，在工具栏处选择“文件”→“新建”→“转换”命令，创建一个转换，名称默认是“转换 1”，具体如图 3-6 和图 3-7 所示。



图 3-6 创建转换

在图 3-7 中选择“文件”→“保存”命令，可以对转换进行重命名以及选择转换保存路径，重命名转换为 example，具体如图 3-8 所示。

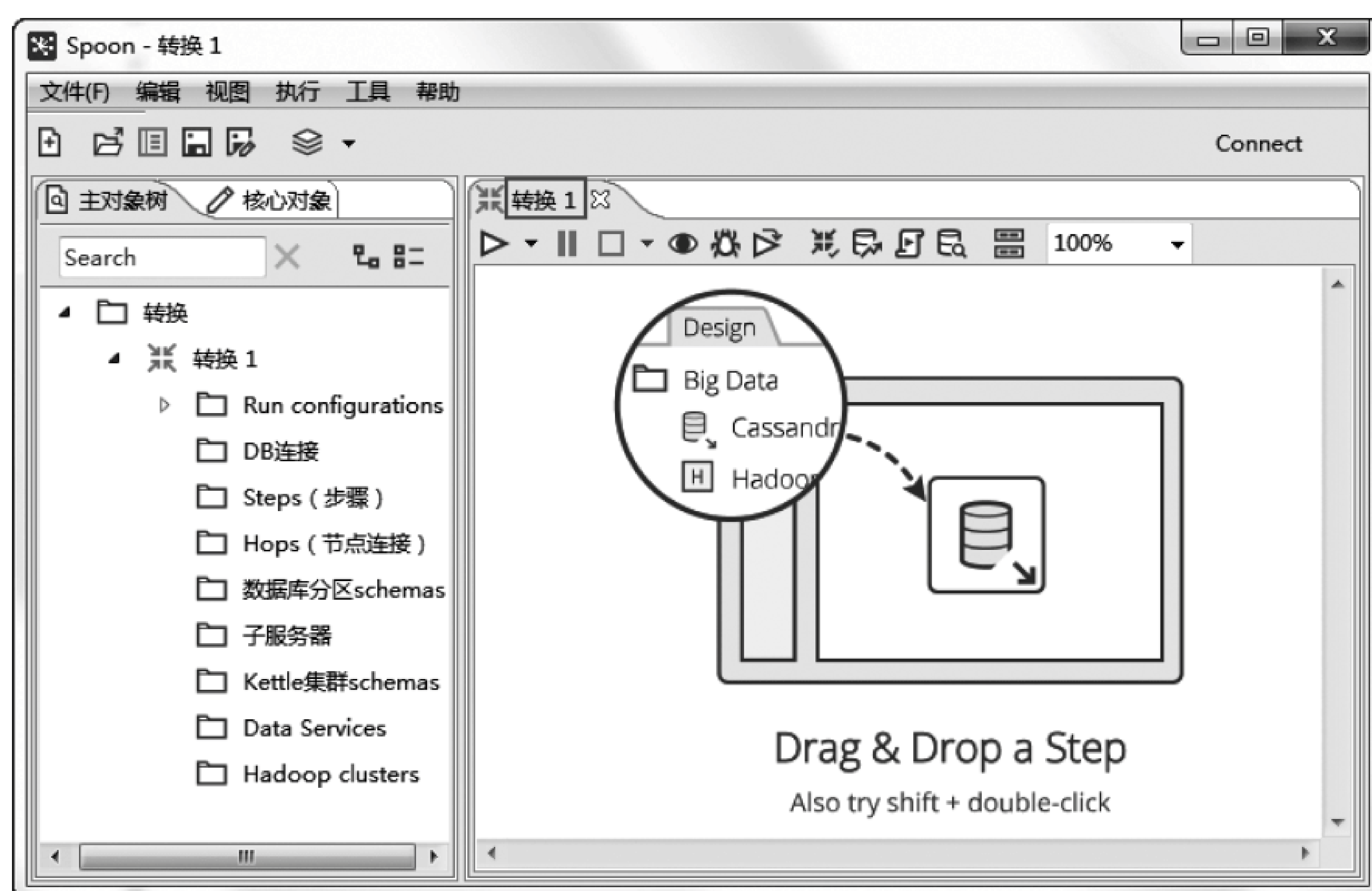


图 3-7　成功创建转换

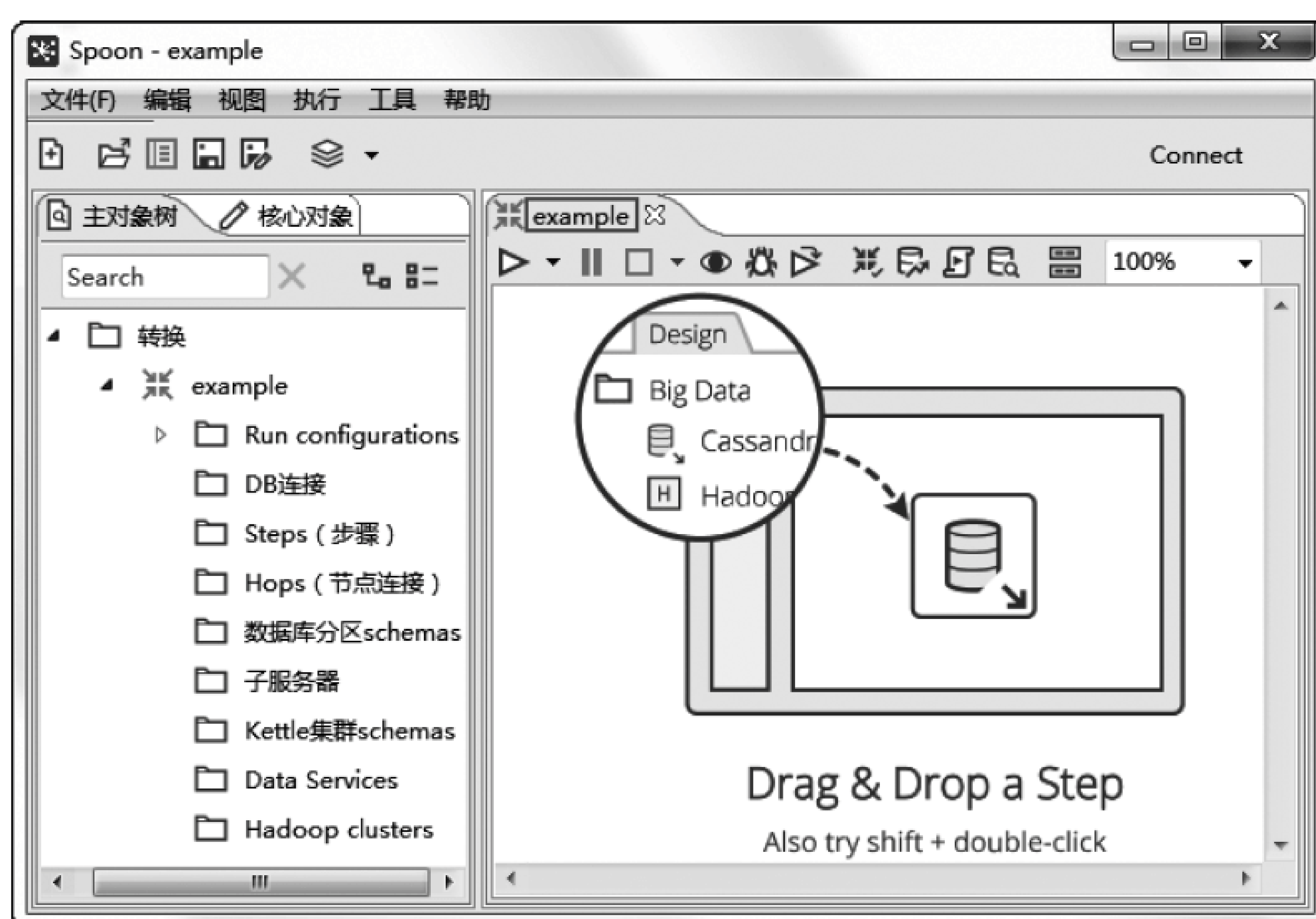


图 3-8　重命名转换为 example

在图 3-8 中，主对象树中的节点主要用于显示当前转换的运行配置参数、数据库连接、步骤以及节点连接(跳)等信息。单击 Kettle 树形列表的核心对象，切换到转换的核心对象界面。转换的核心对象如图 3-9 所示。

从图 3-9 中可以看出，核心对象中包含 Kettle 所有的转换步骤(或转换控件)，后续设计转换操作时，可直接到核心对象中查找所需的转换步骤。

在 Kettle 主界面的工作区右击空白处，从弹出的快捷菜单中选择“新建注释”命令，并添加注释的内容；然后单击“输入”，将“表输入”拖曳到 Kettle 的工作区；单击“输出”，将“文



图 3-9 转换的核心对象

本文件输出”也拖曳到 Kettle 的工作区；按住 Ctrl 键的同时选中“表输入”和“文本文件输出”并右击，从弹出的快捷菜单中选择“新建节点连接”命令，在弹出的窗口中选择“起始步骤”和“目标步骤”，单击“确认”按钮，建立“表输入”向“文本文件输出”的连接，具体效果如图 3-10 所示。



图 3-10 一个简单的转换例子

从图 3-10 中的注释可以看出，这个简单的转换例子是实现从数据库中读取数据，并把数据写到文本文件中，该转换中包含了步骤、跳、注释以及数据行，具体介绍如下。

1. 步骤

步骤是转换里的基本组成部分，也可被称为控件。步骤以图标的方式展现。图 3-10 中显示了两个步骤，即“表输入”步骤和“文本文件输出”步骤。一般地，每个步骤都会具有一些关键特性，具体如下。

- 每个步骤都必须有一个名字，并且这个名字在转换范围内唯一。
- 每个步骤都可以读、写数据行。需要注意的是，“生成记录”步骤除外，因为该步骤只用于写数据。
- 步骤将数据写到与之相连的一个或多个输出跳，再传送到跳的另一端的步骤。对于

另一端的步骤来说，这个跳就是一个输入跳，步骤通过输入跳接收数据。

- 大多数的步骤都可以有多个输出跳，一个步骤的数据发送可以被设置为轮流发送和复制发送。轮流发送是将数据行依次发给每个输出跳。复制发送是将全部数据行发送给所有的输出跳。
- 在运行转换时，一个线程运行一个步骤和步骤的多份副本，所有步骤的线程几乎同时运行，数据行就会连续流过步骤之间的跳。

Kettle 转换中的步骤按功能分类可以分为输入类、输出类、操作类以及脚本类等，每个步骤都完成一种特定的功能。例如，图 3-10 中的“表输入”步骤主要用于向关系型数据库的数据表发出一个 SQL 查询，并将得到的数据行写到它的输出跳中。“文本文件输出”步骤主要用于从它的输入跳读取数据行，并将数据行写到文本文件中。

2. 跳

跳是步骤之间带箭头的连接线，即数据的通道，用于连接两个步骤，实现将元数据从一个步骤传递到另一个步骤，支持分发和复制等方式。这里需要注意的是，由于每个步骤都是单独的线程，当启动转换时，每个步骤都会创建各自的线程并接收和推送传递数据，因此数据处理的顺序并不是按照节点连接箭头的顺序执行的。

实际上，跳是两个步骤之间的被称为行集(Row Set)的数据行缓存(行集的大小可以在转换的设置里定义)。若行集满了，则向行集写数据的步骤将停止写入，直到行集里又有空间。若行集空了，则从行集读取数据的步骤就会停止读取，直到行集里又有可读取的数据行。

跳是基于行集缓存的规则允许每个步骤都由一个独立的线程运行，这样并发程序最高；这一规则也允许数据以最小消耗内存的数据流方式处理。在数据仓库里需要经常处理海量的数据，所以这种并发性高且低耗内存的方式是 ETL 工具的核心需求。

对于 Kettle 来说，不可能定义一个执行顺序，并且也没有必要确定一个起点和终点。因为所有步骤都是以并发方式执行的，当转换启动后，所有步骤都会同时启动，从它们的输入跳中读取数据，并把处理过的数据写到输出跳，直到输入跳里不再有数据就中止步骤的运行；当所有步骤都中止了，那整个转换就中止了。也就是说，从功能角度看，转换有明确的起点和终点。例如，图 3-10 中的转换起点就是“表输入”步骤(该步骤生成数据行)，转换终点是“文本文件输出”步骤(该步骤将数据写到文本文件中，并且后面不再有其他节点)。

需要注意的是，由于转换里的每个步骤都依赖于前一个步骤获取字段值，因此当创建新跳时，在转换里不能进行循环。

3. 注释

注释是一个特殊的存在，不参与程序的处理，它以文本描述的方式呈现在作业中，只为增强流程的可读性，可放在流程图中的任何一个位置。注释的重要性是毋庸置疑的，必要的注释可大大减少维护成本。

4. 数据行

数据是以数据行的形式沿着步骤流动。一个数据行是从零到多个字段的集合，Kettle 中字段的数据类型一共有 10 种，具体见表 3-1。

表 3-1 Kettle 中字段的数据类型

数据类型	相关说明
String	字符类型的数据
Number	双精度浮点数
BigNumber	任意精度数值
Integer	带符号长整型(64 位)
Internet Address	互联网地址
Date	带毫秒精度的日期时间值
Serializable	序列化的数据
Boolean	取值为 true 或 false 的布尔值
Binary	包括图像、声音、视频及其他类型的二进制数据
Timestamp	时间戳

3.3.2 作业

目前,大多数的 ETL 项目都需要完成各种各样的维护工作。例如,如何传送文件、验证数据库中的数据表是否存在等操作,这些操作都必须按照一定顺序完成,由于转换是以并行方式执行的,因此需要一个可以串行执行的作业处理这些操作。

一个作业包含一个或者多个作业项,并且这些作业项都是以某种顺序进行执行的。作业执行的顺序由作业项之间的跳(Job Hop)和每个作业项的执行结果决定。

下面通过一个简单的例子详细讲解 Kettle 中的作业。

双击 Kettle 目录下的 Spoon.bat 脚本,启动 Kettle 的图形化主界面,在工具栏处选择"文件"→"新建"→"作业"命令,创建一个作业,名称默认是"作业 1",具体如图 3-11 和图 3-12 所示。



图 3-11 创建作业

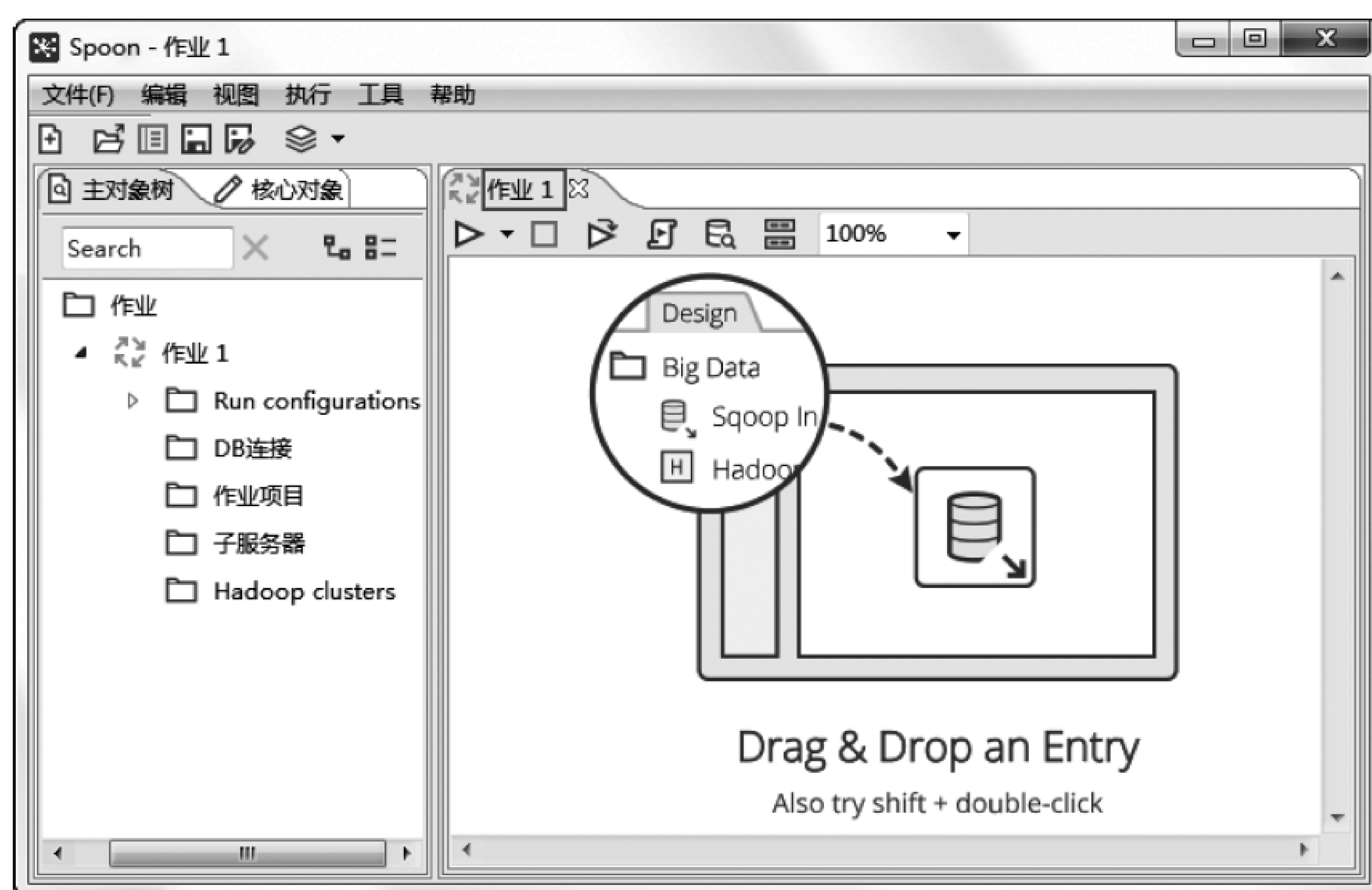


图 3-12　成功创建作业

在图 3-12 中选择“文件”→“保存”命令，可以对作业进行重命名以及选择作业保存路径，作业重命名为 example_job，具体如图 3-13 所示。



图 3-13　作业重命名为 example_job

在图 3-13 中，主对象树中的节点主要用于显示当前作业的运行配置参数、数据库连接以及作业项目等信息。单击 Kettle 树形列表的核心对象，切换到作业的核心对象界面。转换的核心对象如图 3-14 所示。

从图 3-14 中可以看出，作业核心对象中包含 Kettle 所有作业的作业项（或作业控件），后续设计作业操作时，可直接到作业核心对象中查找所需的作业项。

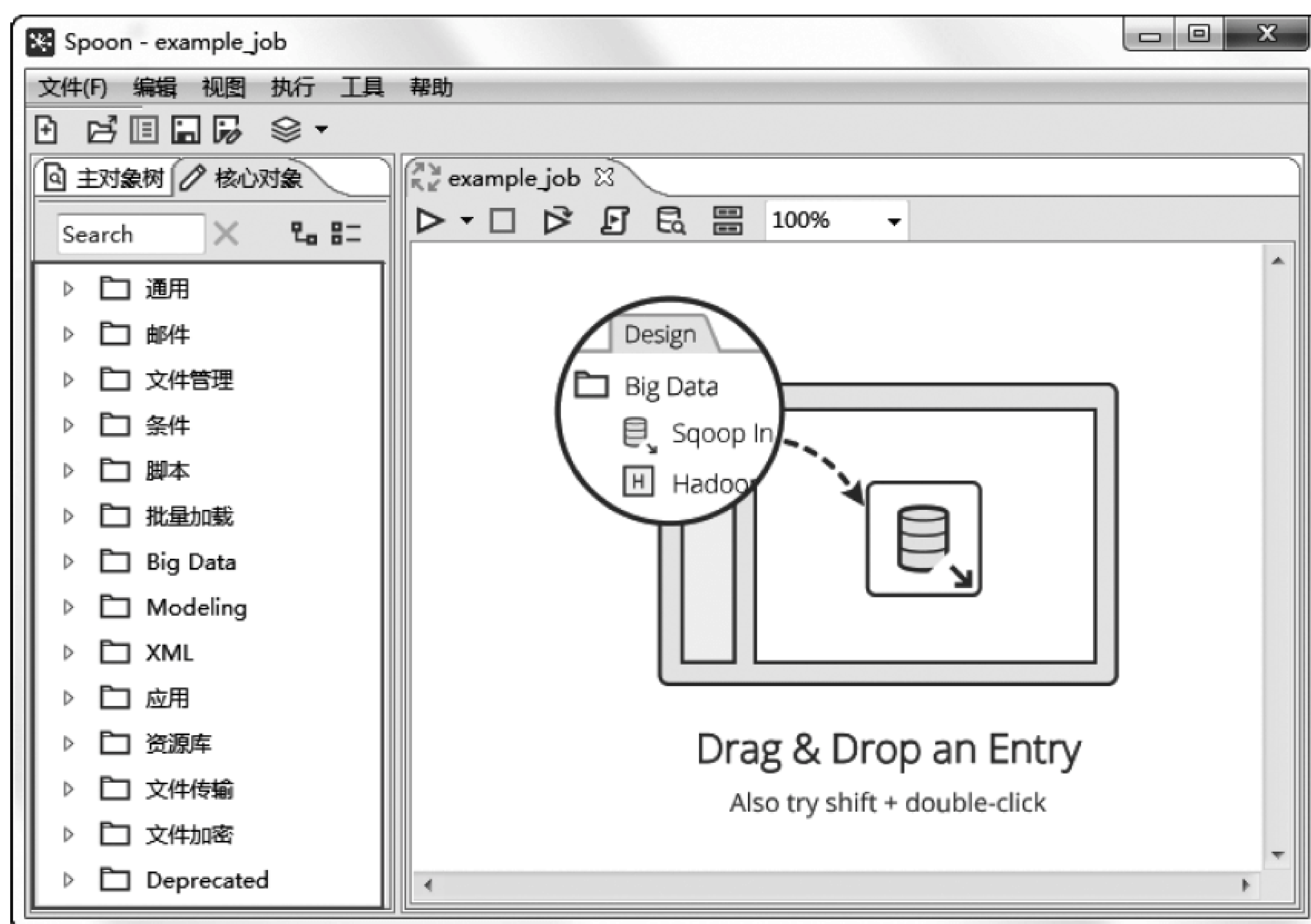


图 3-14 转换的核心对象

在 Kettle 主界面的工作区右击空白处，从弹出的快捷菜单中选择“新建记录”命令，并添加注释的内容；然后单击“通用”，将 Start 和“作业”依次拖曳到 Kettle 的工作区；单击“邮件”，将“发送邮件”也拖曳到 Kettle 的工作区；然后同时选中 Start 和“作业”右击，从弹出的快捷菜单中选择“新节点”命令，建立 Start 和“作业”之间的连接，再通过同样的操作将“作业”与“作业”、“作业”与“发送邮件”之间也建立连接，具体效果如图 3-15 所示。

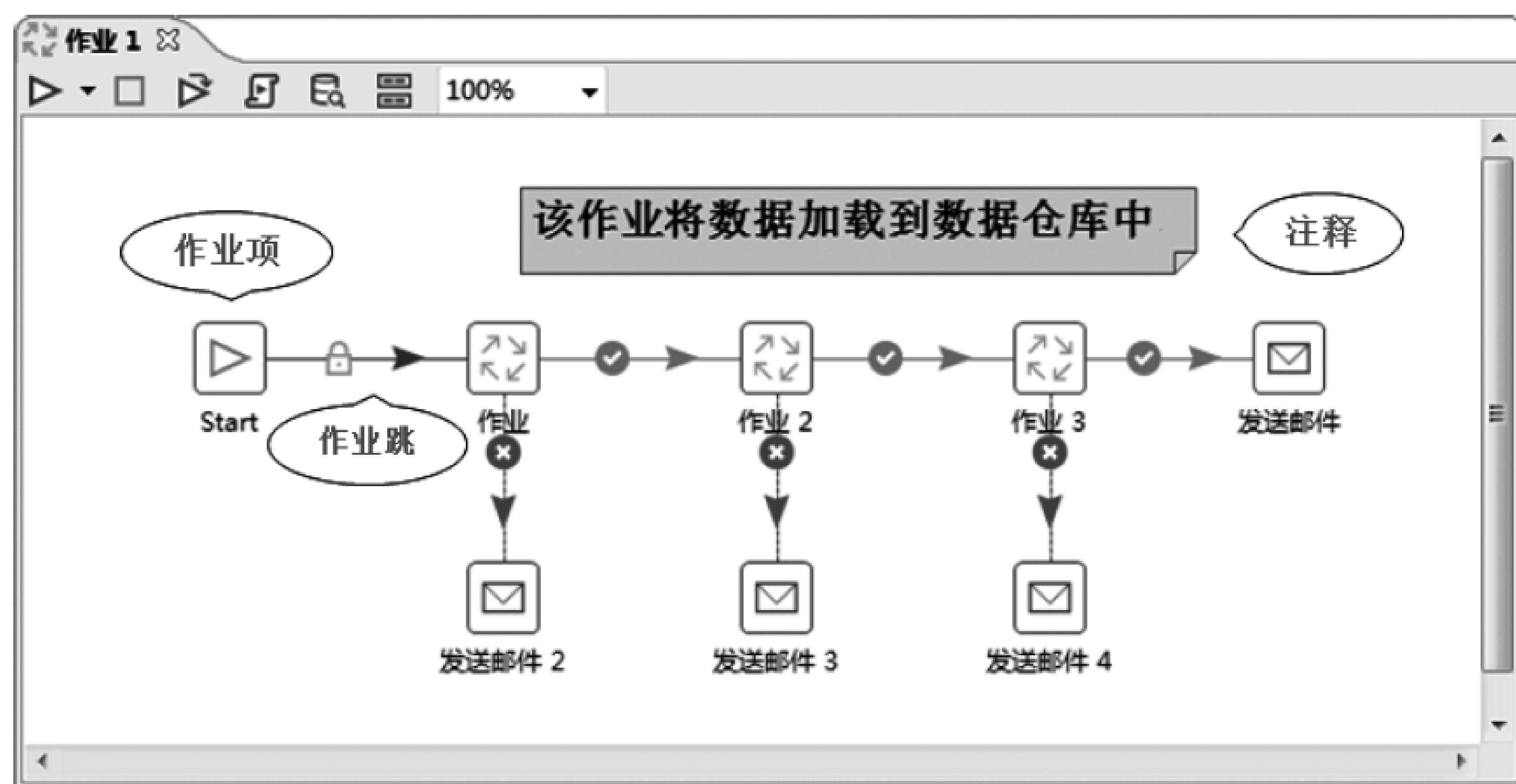


图 3-15 一个简单的作业例子

为了便于理解，对图 3-15 中的作业项进行重命名，具体如图 3-16 所示。

从图 3-16 中可以看出，该作业是一个典型的加载数据到数据仓库的作业，该作业中包含作业项、作业跳以及多路径和回溯，具体介绍如下。

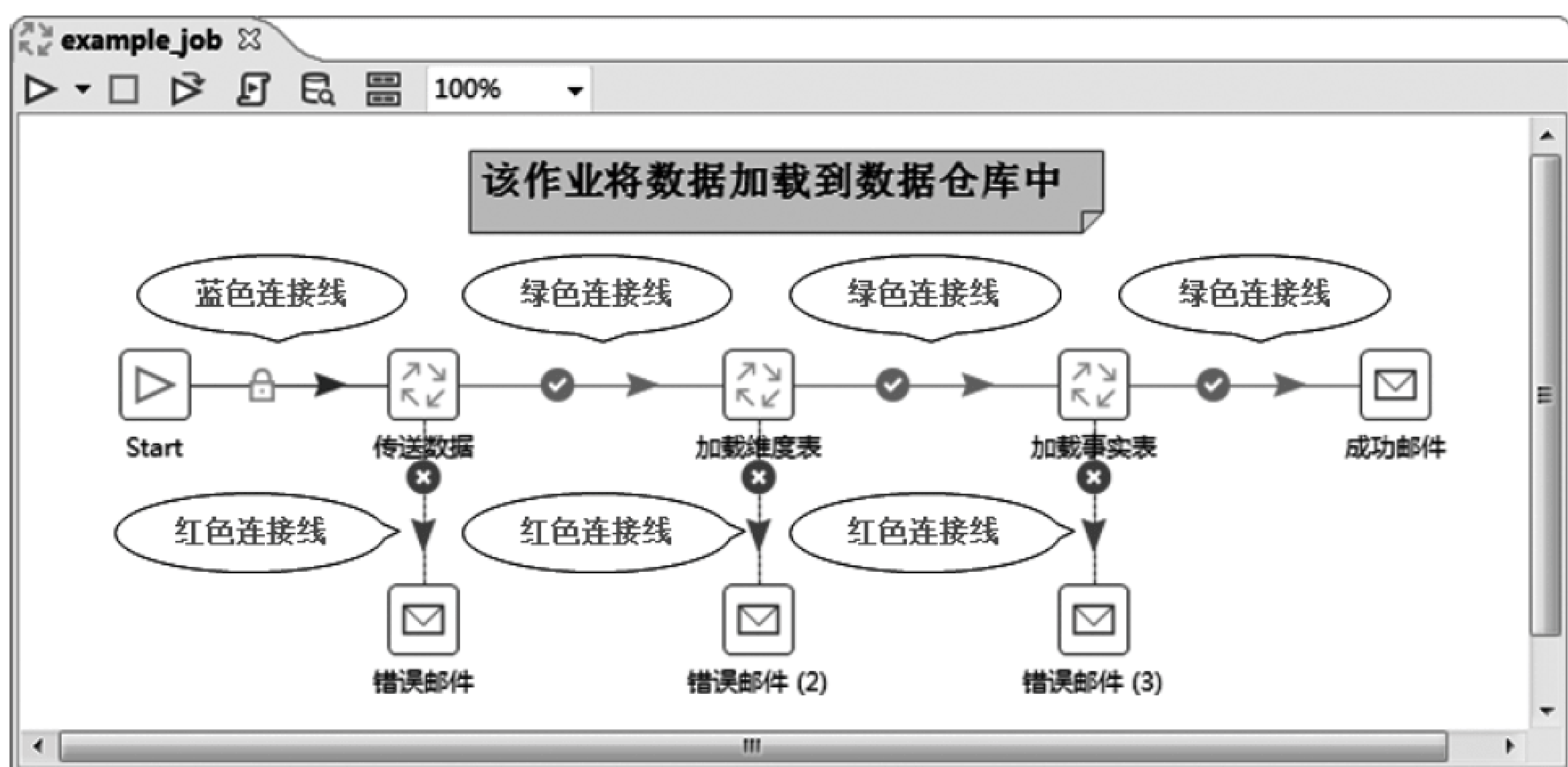


图 3-16　对作业项进行重命名

1. 作业项

作业项是作业的基本构成部分，也可称为控件。作业项类似于转换中的步骤，也可以使用图标的方式进行图形化展示。作业项与步骤有很大的区别，具体如下。

- 步骤的名字是唯一的，而作业项可以进行复制。也就是说，可以将一个作业项放在多个不同的位置，并且这些复制的作业项中的信息都是相同的，若修改了其中一个作业项，那么其他复制的作业项也都会随之修改。
- 步骤之间的数据行是以数据流的方式传递的，而作业项之间可以传递一个结果对象，并且结果对象里包含了数据行，但数据行不是以流的方式传递，而是等到一个作业项执行完成后，再传递给下一个作业项。
- 默认情况下，所有步骤都是以并行的方式执行，而所有作业项目都是串行方式执行的。

2. 作业跳

作业跳是作业项之间的连接线，它定义了作业的执行路径。作业里每个作业项的不同运行结果决定了作业的不同执行路径，具体如下。

- 无条件执行：不论上一个作业项执行成功，还是失败，下一个作业项都会执行，如图 3-16 中的蓝色连接线，并且上面有一个锁的图标。
- 当运行结果为“真”时，则执行：当上一个作业项的执行结果为“真”时，执行下一个作业项。通常在需要无错误执行的情况下使用，如图 3-16 中的绿色连接线，并且上面有一个对钩的图标。
- 当运行结果为“假”时，则执行：当上一个作业项的执行结果为“假”或者没有成功执行时，执行下一个作业项，如图 3-16 中的红色连接线，并且上面有一个红色的停止图标。

3. 多路径和回溯

Kettle 使用一种回溯算法执行作业里的所有作业项，并且作业项的执行结果(真/假)决定执行的路径。回溯算法：假设执行到一条路径的某个节点时，依次执行这个节点的所有子路径，直到没有可执行的子路径，就返回该节点的上一个节点，如此反复。

下面通过一个简单的例子介绍回溯算法，具体如图 3-17 所示。

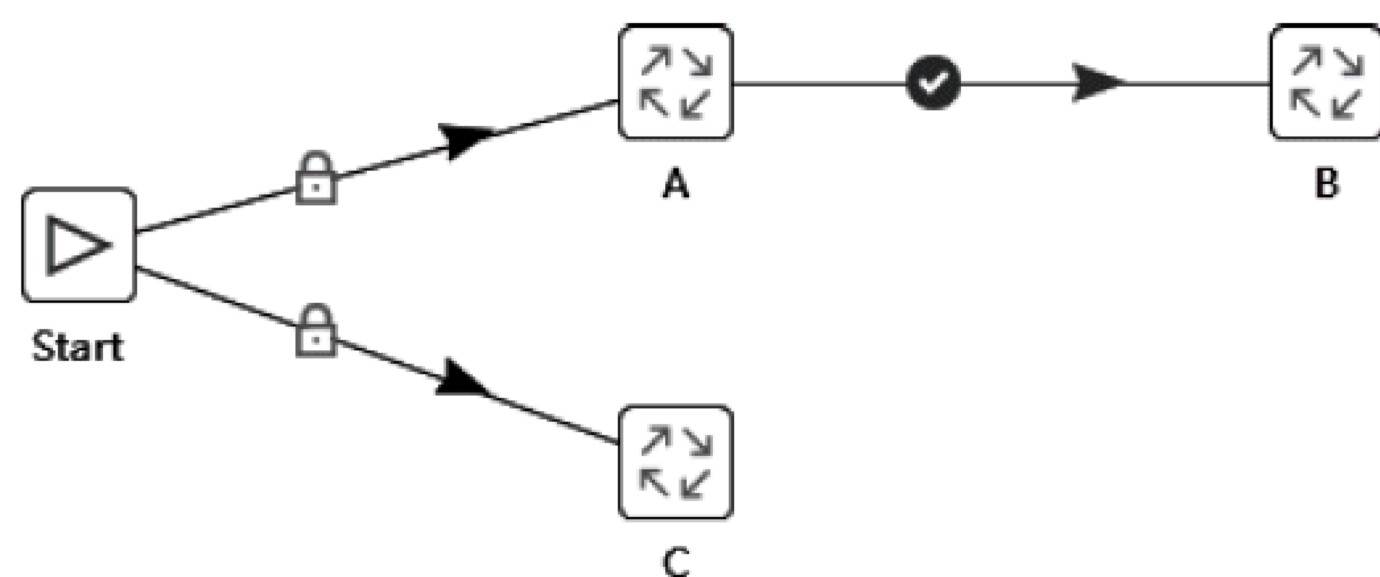


图 3-17 使用回溯算法串行执行多个路径

在图 3-17 中，作业项 A、B、C 的执行顺序具体如下。

(1) Start 作业项搜索下一个节点的所有作业项，即作业项 A 和 C。

(2) 执行 A 作业项。

(3) 搜索 A 作业项后面的作业项，即作业项 B。

(4) 执行 B 作业项。

(5) 搜索 B 作业项后面的作业项，没有找到任何作业项。

(6) 回到 A 作业项，也没有找到其他作业项。

(7) 回到 Start 作业项，发现另一个要执行的作业项 C。

(8) 执行 C 作业项。

(9) 搜索 C 作业项后面的作业项，没有找到任何作业项。

(10) 回到 Start 作业项，没有找到其他作业项。

(11) 作业结束。

由于没有定义作业的执行顺序，上述执行顺序为作业项 A→B→C，上述执行顺序也可以为作业项 C→A→B。

4. 作业项结果

作业项的执行结果不仅决定了作业的执行路径，还向下一个作业项传递了一个结果对象，结果对象包含一组数据行、一组文件名、行数(读、写、输入、输出、更新、删除、拒绝的行数)、错误数(转换中的错误数)以及脚本作业项的退出状态。

3.4 Kettle 的基本功能

3.4.1 转换管理

在 Kettle 工具中，转换管理主要包括输入、输出、转换、应用、流程、脚本、查询、连接、检验、作业、映射、批量加载等功能。下面通过一张表描述 Kettle 转换功能常用的控件，具体

见表 3-2。

表 3-2　Kettle 转换功能常用的控件

转换类别	步骤/控件	相 关 说 明
输入	CSV 文件输入	从本地的 CSV 文件中输入数据
	文本文件输入	从本地的文本文件中输入数据
	表输入	从数据库的数据表中输入数据
	获取系统信息	读取系统信息输入数据
输出	文本文件输出	将处理后的结果输出到文本文件中
	表输出	将处理后的结果输出到数据库的数据表中
	插入/更新	根据处理后的结果对数据库中的数据表进行插入更新。根据查询条件中的字段判断数据表中是否存在相关记录,若存在,则进行插入,否则进行更新
转换	值映射	数据的映射
	列转行	将数据表的列转成数据表的行
	去除重复记录	从输入流中去除重复的数据,需要注意的是输入流中的数据必须是已排序的
	唯一行(哈希值)	从输入流中去除重复的数据,不需要对输入流中的数据进行排序
	字段选择	选择需要的字段,过滤掉不要的字段,也可与数据库字段对应
	拆分字段	将一个字段拆分成多个字段
	排序记录	基于某个字段值将数据进行升序或降序处理
	行转列	将数据表的行转成数据表的列
	增加常量	增加需要的常量字段
应用	替换 NULL 值	若某个字符串的值为 NULL,则指定某个字符串的值进行替换
	设置值为 NULL	若某个字符串的值等于指定的值,则将这个字符串的值设置为空
流程	空操作	不做任何操作,一般充当一个占位符
	过滤记录	根据条件对数据进行过滤分类
脚本	Java 代码	转换的扩展功能,编写 Java 脚本,对数据进行相应的处理
	JavaScript 代码	转换的扩展功能,编写 JavaScript 脚本,对数据进行相应的处理
	执行 SQL 脚本	执行 SQL 脚本,对数据进行相应的处理
查询	HTTP Client	通过一个可以动态设定参数的基本网址调用 HTTP Web 服务
	流查询	将目标表读取到内存,通过查询条件对内存中的数据集进行查询
	数据库查询	根据设定的查询条件对目标表进行查询,返回需要的结果字段
连接	合并记录	合并两个数据流,并根据某个关键字排序
	排序合并	合并多个数据流,并且数据的行要基于某个关键字进行排序

续表

转换类别	步骤/控件	相关说明
作业	复制记录到结果	将数据写入正在执行的任务中
	获取变量	获取环境或 Kettle 变量
	设置变量	设置环境变量

表 3-2 中列举了一些 Kettle 转换功能常用的控件。下面通过 Kettle 工具的转换实现将一张数据表中的两个字段进行拼接,然后插入到另一张数据表中。

1. 数据准备

创建一个数据库 personal,并在该数据库中创建两张数据表,即数据表 personal_a 和数据表 personal_b(数据表的创建过程,这里不再赘述)。数据表 personal_a 和数据表 personal_b 分别如图 3-18 和图 3-19 所示。

1 信息 | 2 表数据 | 3 信息

限制行 第一行: 0

id	surname	name	age	sex
p001	张	三	18	male
p002	李	四	19	female
p003	王	五	18	female
p004	赵	六	20	female
p005	孙	七	19	male
p006	周	八	21	female
p007	吴	九	20	male

数据库: personal 表格: personal_a

图 3-18 数据表 personal_a

1 信息 | 2 表数据 | 3 信息

限制行 第一行: 0

id	username	age	sex
(NULL)	(NULL)	(NULL)	(NULL)

数据库: personal 表格: personal_b

图 3-19 数据表 personal_b

2. 打开 Kettle 工具,创建转换

通过使用 Kettle 工具创建一个转换 field_stitching,并添加“表输入”控件、“JavaScript 代码”控件、“插入/更新”控件以及跳连接线,具体效果如图 3-20 所示。

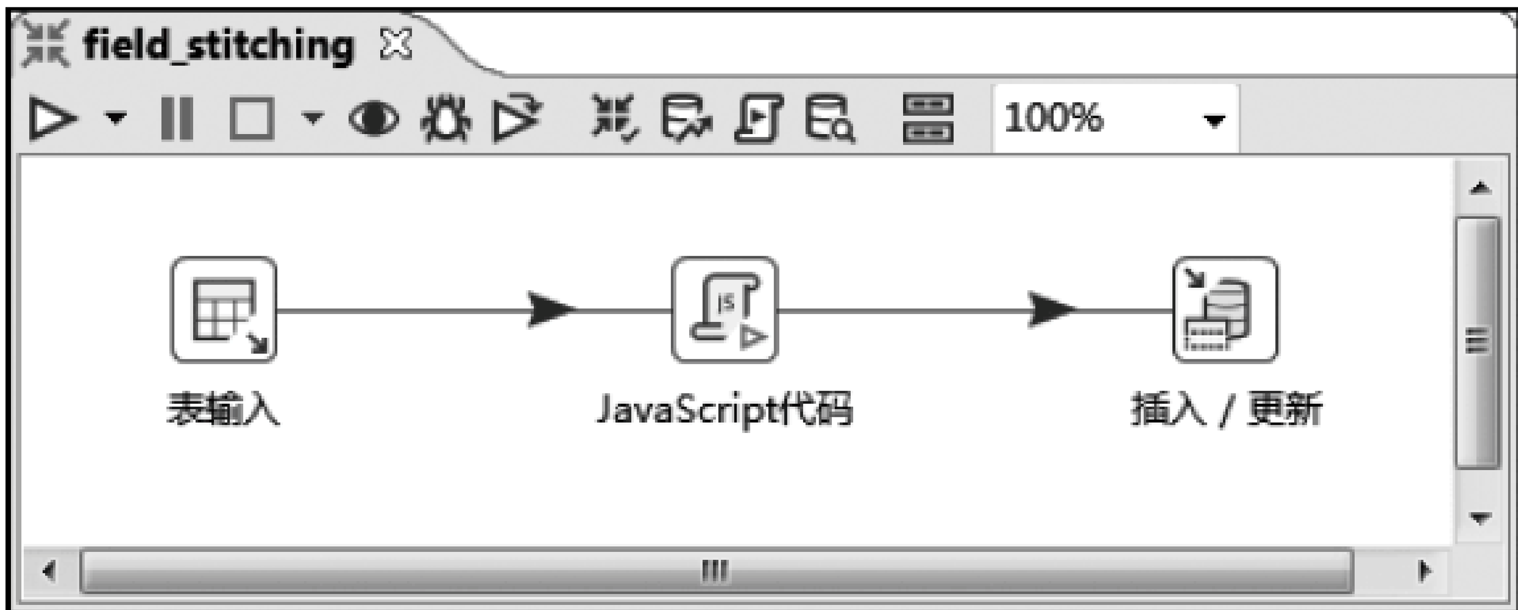


图 3-20 创建转换 field_stitching

3. 配置"表输入"控件

双击图 3-20 中的"表输入"控件,进入"表输入"界面,具体如图 3-21 所示。



图 3-21 "表输入"界面

单击图 3-21 中的"新建"按钮,配置数据库连接,配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 3-22 所示。



图 3-22 MySQL 数据库连接的配置

单击图 3-21 中的"获取 SQL 查询语句..."按钮,弹出"数据库浏览器"窗口,展开 field_stitching,并选中"表"菜单下的数据表 personal_a,具体如图 3-23 所示。

单击图 3-23 中的"确定"按钮,弹出"问题?"窗口,具体如图 3-24 所示。



图 3-23　选择数据表 personal_a

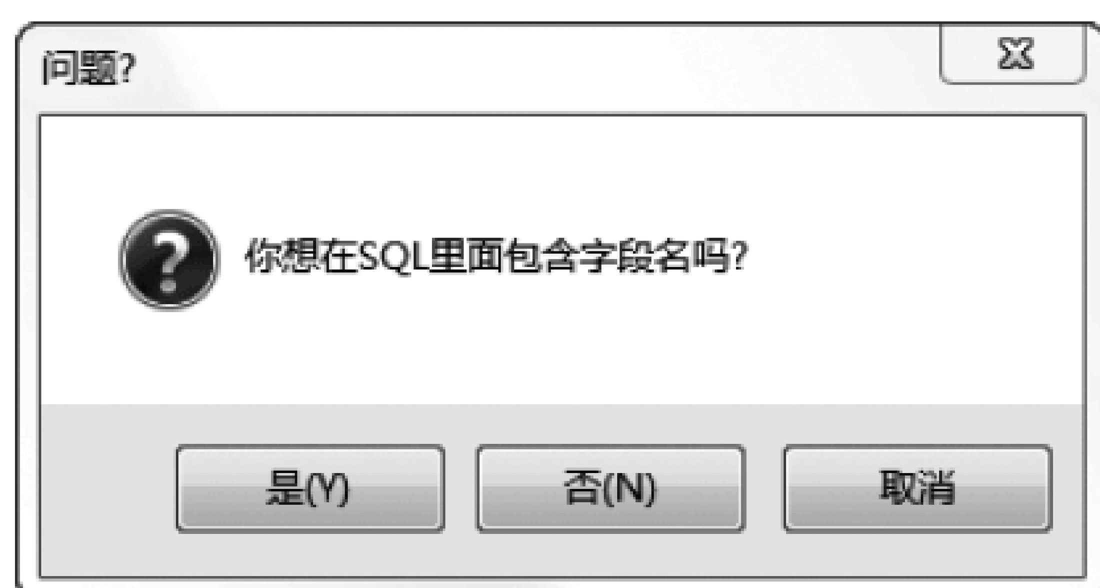


图 3-24　选择是否在 SQL 里面包含字段名

单击图 3-24 中的“是”按钮,“表输入”界面配置的最终效果如图 3-25 所示。



图 3-25　“表输入”界面配置的最终效果

单击图 3-25 中的“确定”按钮,完成“表输入”控件的配置。

4. 配置“JavaScript 代码”控件

双击图 3-20 中的“JavaScript 代码”控件,进入“JavaScript 代码”界面,具体如图 3-26 所示。



图 3-26　"JavaScript 代码"界面

在图 3-26 的 JavaScript 代码窗口中编写 JavaScript 脚本代码，然后单击"获取变量"按钮，在字段窗口的"改名为"字段处添加新的字段名称 username，具体如图 3-27 所示。



图 3-27　配置"JavaScript 代码"控件

单击图 3-27 中的"确定"按钮，完成"JavaScript 代码"控件的配置。

5. 配置"插入/更新"控件

双击图 3-20 中的"插入/更新"控件，进入"插入/更新"界面，如图 3-28 所示。



图 3-28 “插入/更新”界面

单击图 3-28 中的“新建”按钮,配置数据库连接,配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 3-29 所示。



图 3-29 MySQL 数据库连接的配置

单击图 3-28 中目标表右侧的“浏览”按钮,弹出“数据库浏览器”窗口,展开 field_stitching1,并选中“表”菜单下的数据表 personal_b,具体如图 3-30 所示。



图 3-30 选择要插入数据的数据表 personal_b

单击图 3-30 中的“确定”按钮，完成目标表的选择，具体效果如图 3-31 所示。



图 3-31 目标表 personal_b

单击图 3-31 中的“获取字段”按钮，用来指定查询数据需要的关键字，这里选择的是数据表 personal_b 中的 id_b 字段和输入流里的 id 字段，具体如图 3-32 所示。

单击图 3-32 中的“编辑映射”按钮，弹出“映射匹配”窗口，具体如图 3-33 所示。

选中图 3-33 中的“源字段”选项框中的字段和“目标字段”选项框中的字段，然后单击 Add 按钮，依次将一对对映射字段添加至“映射”选项框中，若“源字段”选项框的字段和“目标字段”选项框的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 3-34 所示。

单击图 3-34 中的“确定”按钮，完成源字段与目标字段的映射匹配，效果如图 3-35 所示。



图 3-32 获取字段



图 3-33 “映射匹配”窗口



图 3-34 配置源字段与目标字段的映射匹配



图 3-35　字段映射匹配的效果

单击图 3-35 中的“确定”按钮，完成“插入/更新”控件的配置。

6. 运行转换 field_stitching

单击转换工作区顶部的 ▷ 按钮，运行创建的转换 field_stitching，实现将数据表 personal_a 中的 surname 字段和 name 字段进行拼接，并将结果数据插入到数据表 personal_b 中，具体如图 3-36 所示。

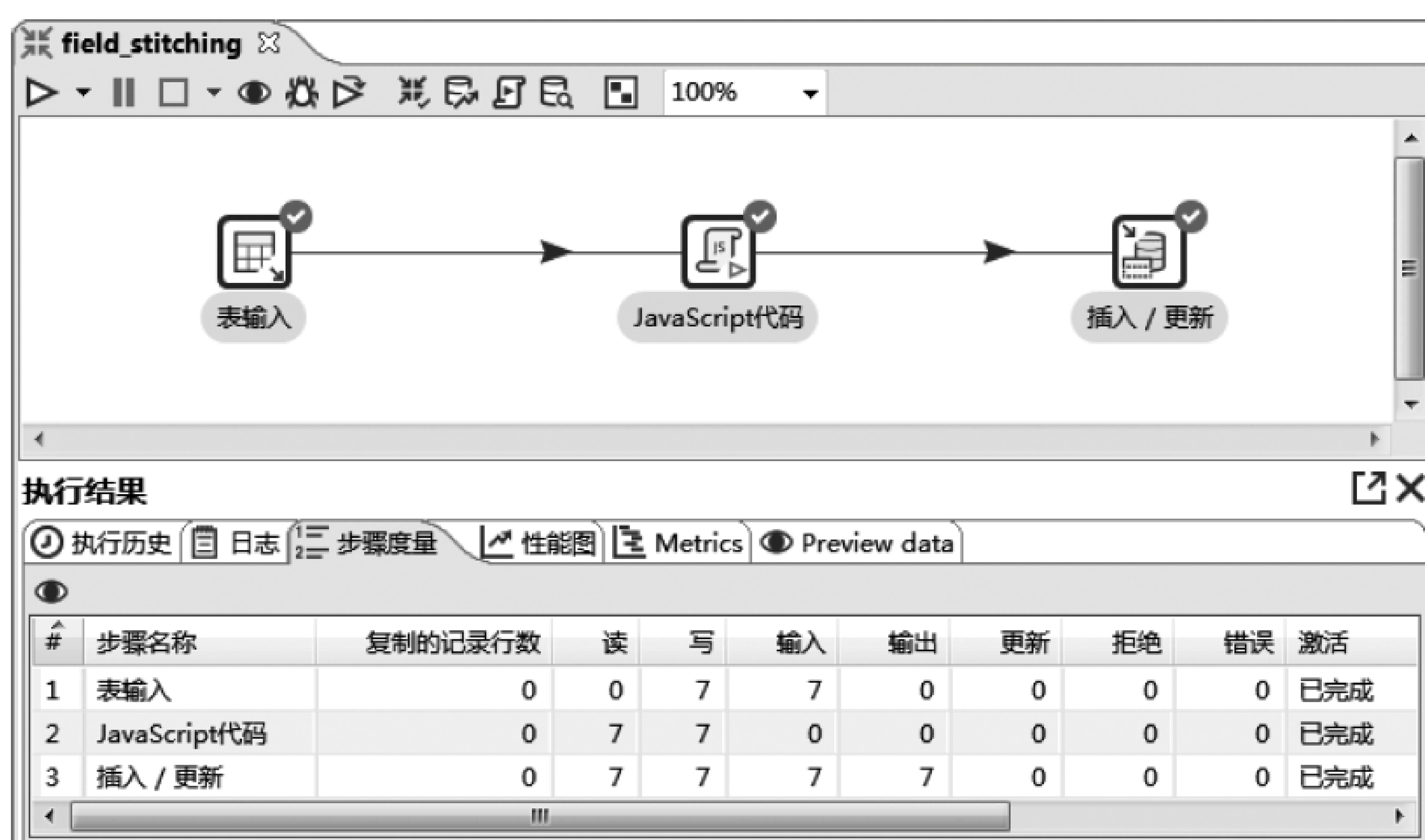


图 3-36　运行转换 field_stitching

从图 3-36 中执行结果一栏的“步骤度量”选项卡中可以看出，“表输入”控件执行 7 行数据的写入和输入操作；“JavaScript 代码”控件执行读、写操作各 7 行数据；“插入/更新”控件执行读、写以及输入、输出操作各 7 行数据。也就是说，“插入/更新”控件将从“JavaScript 代码”控件的输入流中读入的 7 条数据均写入到数据表 personal_b 中。

7. 查看数据表 personal_b 中的数据

通过 SQLyog 工具，查看数据表 personal_b 是否已成功插入 7 条数据，结果如图 3-37 所示。

1 信息　2 表数据　3 信息

限制行 第一行: 0

id_b	username	age	sex
p001	张三	18	male
p002	李四	19	female
p003	王五	18	female
p004	赵六	20	female
p005	孙七	19	male
p006	周八	21	female
p007	吴九	20	male

数据库: personal 表格: personal_b

图 3-37　数据表 personal_b

从图 3-37 中可以看出，数据表 personal_b 中已插入了数据，并且 username 字段显示的是数据表 personal_a 中 surname 字段和 name 字段进行拼接后的结果数据，因此可以说明我们成功实现了将数据表 personal_a 中的 surname 字段和 name 字段进行拼接，并将结果数据插入到数据表 personal_b 中。

3.4.2　作业管理

在 Kettle 工具中，作业管理主要包括通用、邮件、文件管理、条件、脚本、批量加载等功能。下面通过一张表描述 Kettle 作业功能常用的控件，具体见表 3-3。

表 3-3 中列举了一些 Kettle 作业功能常用的控件。下面通过 Kettle 工具的作业发送邮件。

表 3-3　Kettle 作业功能常用的控件

作业类别	步骤/控件	相关说明
通用	Start	作业执行的开始
	Dummy	作业执行的结束
	作业	使用新的作业执行之前已定义好的作业
	成功	提示作业执行成功
	转换	使用作业执行之前已定义好的转换流程
邮件	POP 收信	通过设置好的 POP 服务器地址发送邮件
	发送邮件	发送作业执行成功或失败邮件

续表

作业类别	步骤/控件	相关说明
文件管理	创建文件	创建一个新的文件，若文件名已存在，则提示创建失败并退出
	删除一个文件	删除指定文件名的文件，若不存在指定的文件名称，则提示删除失败
	复制文件	将源文件中的内容复制到新创建的文件中或替换已存在的文件
	比较文件	比较两个文件中的内容
	移动文件	将文件移动到另一个文件夹中
	解压缩文件	将作业文件进行解压或压缩操作
条件	检查表是否存在	检查数据库中的数据表是否存在
	检查一个文件是否存在	检查指定的文件是否存在
脚本	JavaScript	编写 JavaScript 脚本，进行相应的数据处理
	Shell	编写 Shell 脚本，进行相应的数据处理
	SQL	编写 SQL 脚本，进行相应的数据处理
批量加载	MySQL 批量加载	将本地文件中的数据批量加载到 MySQL 数据库中
	SQL Server 批量加载	将本地文件中的数据批量加载到 SQL Server 数据库中
	从 MySQL 批量导出到文件	将 MySQL 数据库中的数据批量导出到本地文件中

1．打开 Kettle 工具，创建作业

通过使用 Kettle 工具，创建一个作业 send_email，并添加 Start 控件、“发送邮件”控件、“成功”控件以及作业跳连接线，具体效果如图 3-38 所示。

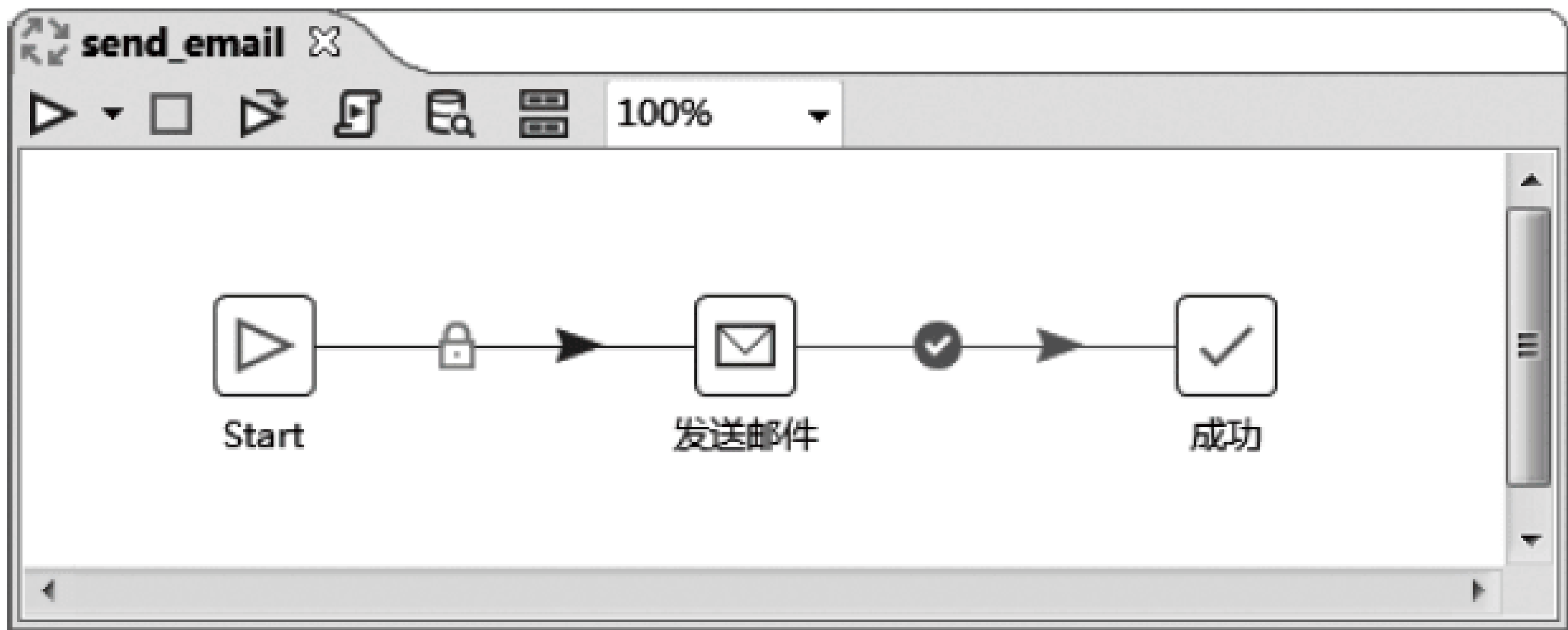


图 3-38　创建作业 send_email

2．配置 Start 控件

双击图 3-38 中的 Start 控件，进入“作业定时调度”界面，具体如图 3-39 所示。

单击图 3-39 中“类型”后的下拉列表框，选择“时间间隔”定时，并设置以秒计算的间隔是 10，以分钟计算的间隔是 0（即作业运行 10 秒后发送邮件），具体如图 3-40 所示。



图 3-39 “作业定时调度”界面



图 3-40 设置发送邮件的时间间隔

单击图 3-40 中的“确定”按钮，完成 Start 控件的配置。

3. 配置“发送邮件”控件

双击图 3-38 中的“发送邮件”控件，进入“发送邮件”界面，具体如图 3-41 所示。



图 3-41 “发送邮件”界面

在图 3-41 中添加收件人和发件人的相关信息，这里添加了收件人地址、发件人的回复名称以及发件人地址，具体如图 3-42 所示。

单击图 3-42 中的“服务器”选项卡，具体如图 3-43 所示。

在图 3-43 中添加邮件服务器的相关参数和验证发件人的信息，具体如图 3-44 所示。

在图 3-44 中，邮件服务器与发件人的账号对应，由于发件人的邮件是 263 企业邮箱，所以 SMTP 服务器和端口号分别填写 smtp. 263. net 和 25。若所填发件人的邮箱是 QQ 邮箱，则填写 smtp. qq. com 和 465（或 587）；若所填发件人的邮箱是网易 163 邮箱，则填写 smtp. 163. com 和 25；若所填发件人的邮箱是其他邮箱，则自行查阅填写，这里不再赘述。

图 3-42　添加收件人、发件人相关信息

图 3-43　“服务器”选项卡

图 3-44　配置邮件服务器

单击图 3-44 中的“邮件消息”选项卡，具体如图 3-45 所示。



图 3-45 “邮件消息”选项卡

在图 3-45 中勾选“信息里带日期?”和“使用 HTML 邮件格式?”复选框，设置发送的邮件信息里带日期，并且发送的邮件使用的是 HTML 格式；在“消息”框中添加邮件的主题和注释，具体如图 3-46 所示。



图 3-46 配置邮件消息

在图 3-46 中单击“确定”按钮，完成“发送邮件”控件的配置。

4. 运行 send_email 作业

单击作业工作区顶部的▷按钮，运行创建的 send_email 作业，实现发送邮件的功能，具体如图 3-47 所示。

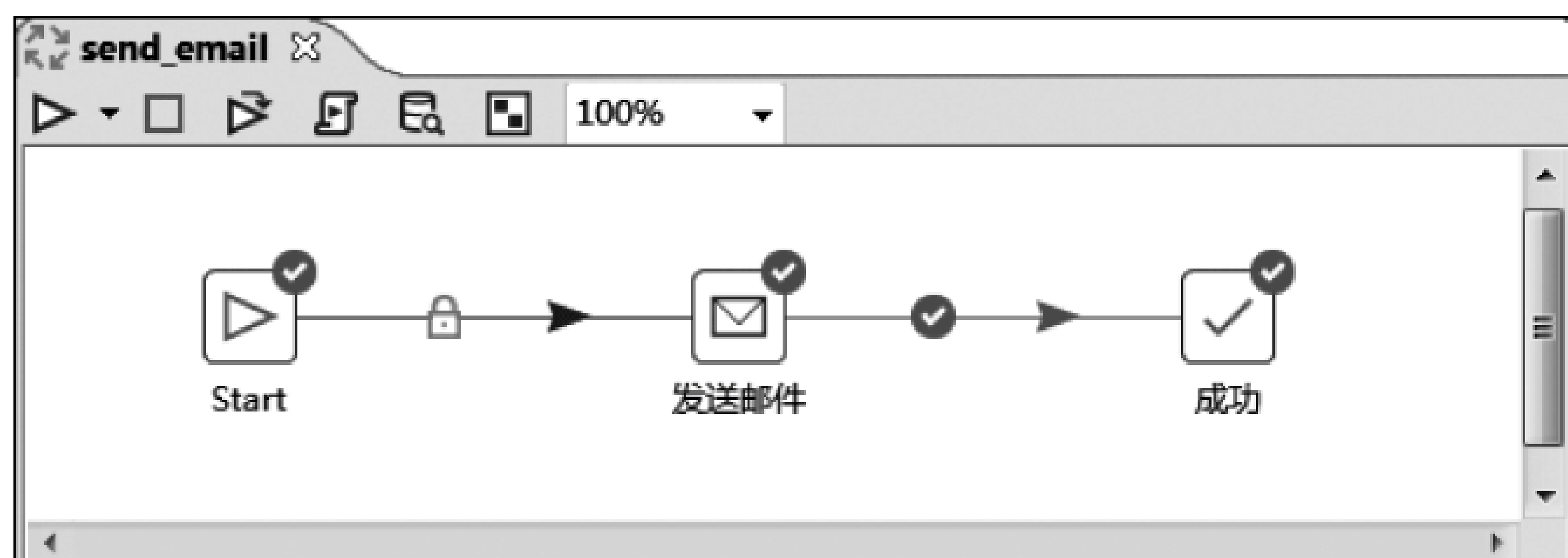
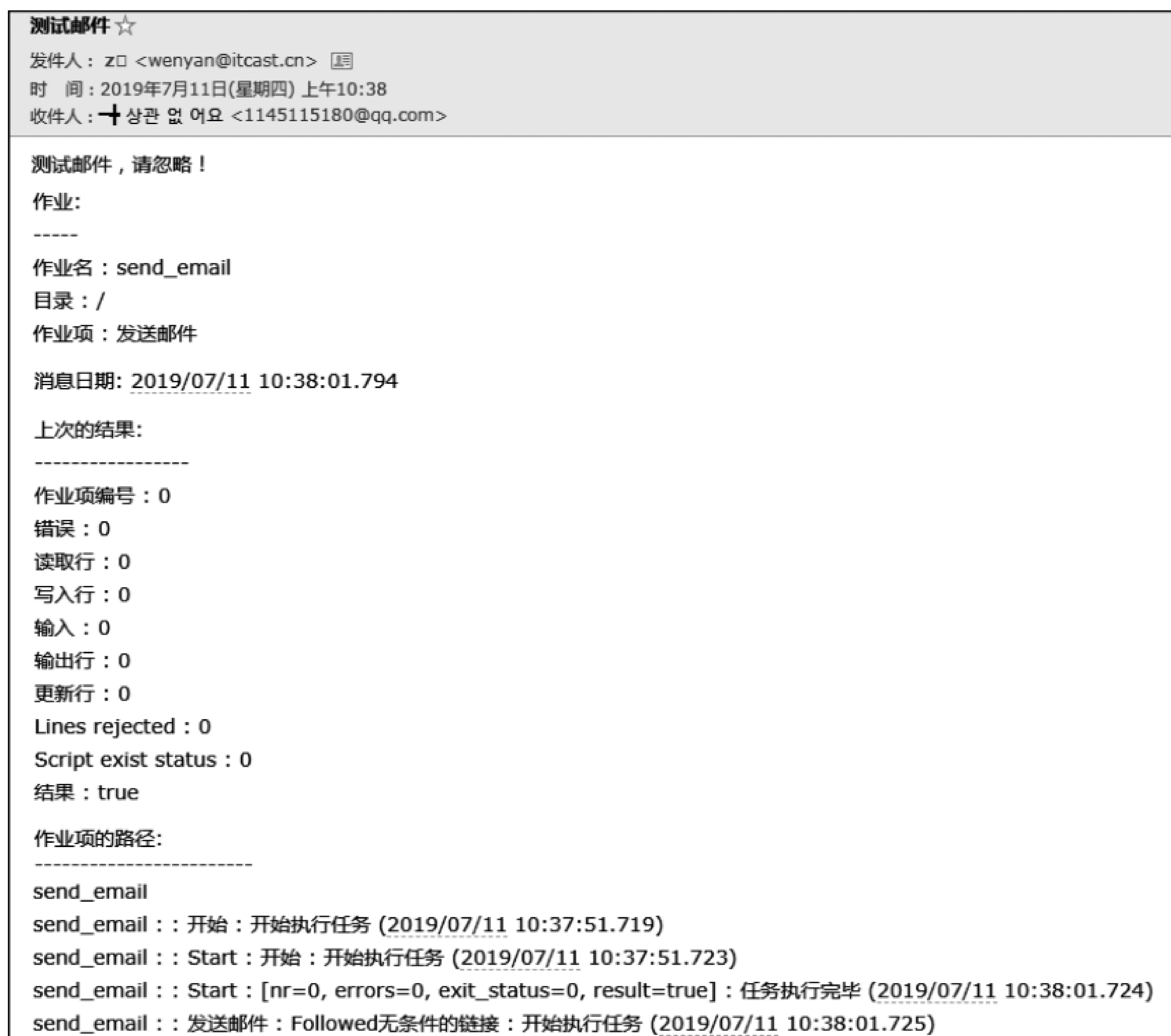


图 3-47　运行 send_email 作业

从图 3-47 中各控件右上角的绿色对勾图标可以看出，send_email 作业运行成功。

5. 查看邮箱的收件箱

通过查看 QQ 邮箱的邮件，验证是否收到发送的邮件，具体如图 3-48 所示。



测试邮件 ☆

发件人： z□ <wenyan@itcast.cn>

时　间：2019年7月11日(星期四) 上午10:38

收件人：━╋ 상관 없 어요 <1145115180@qq.com>

测试邮件，请忽略！

作业:

作业名：send_email

目录：/

作业项：发送邮件

消息日期: 2019/07/11 10:38:01.794

上次的结果:

作业项编号：0

错误：0

读取行：0

写入行：0

输入：0

输出行：0

更新行：0

Lines rejected：0

Script exist status：0

结果：true

作业项的路径:

----

send_email

send_email：：开始：开始执行任务 (2019/07/11 10:37:51.719)

send_email：：Start：开始：开始执行任务 (2019/07/11 10:37:51.723)

send_email：：Start：[nr=0, errors=0, exit_status=0, result=true]：任务执行完毕 (2019/07/11 10:38:01.724)

send_email：：发送邮件：Followed无条件的链接：开始执行任务 (2019/07/11 10:38:01.725)

图 3-48　邮件内容

从图 3-48 中可以看出，收到了发件人 wenyan@itcast.cn 的邮件，并且邮件内容包含主

题、注释以及与作业相关的信息，因此可以说明我们通过 Kettle 工具的作业实现了发送邮件的功能。

3.4.3 数据库连接

前面介绍了 Kettle 工具中的转换管理和作业管理，其中转换管理中使用数据库连接获取数据库数据，而 Kettle 中的数据库连接实际上是数据库连接的描述，也就是实际建立数据库连接需要的参数，实际数据库连接只在运行时才会创建，因此，定义一个 Kettle 的数据库连接并不会真正打开一个数据库连接。

由于数据库的种类有很多，因此 Kettle 工具的数据库连接窗口包含多种数据库类型，具体如图 3-49 所示。

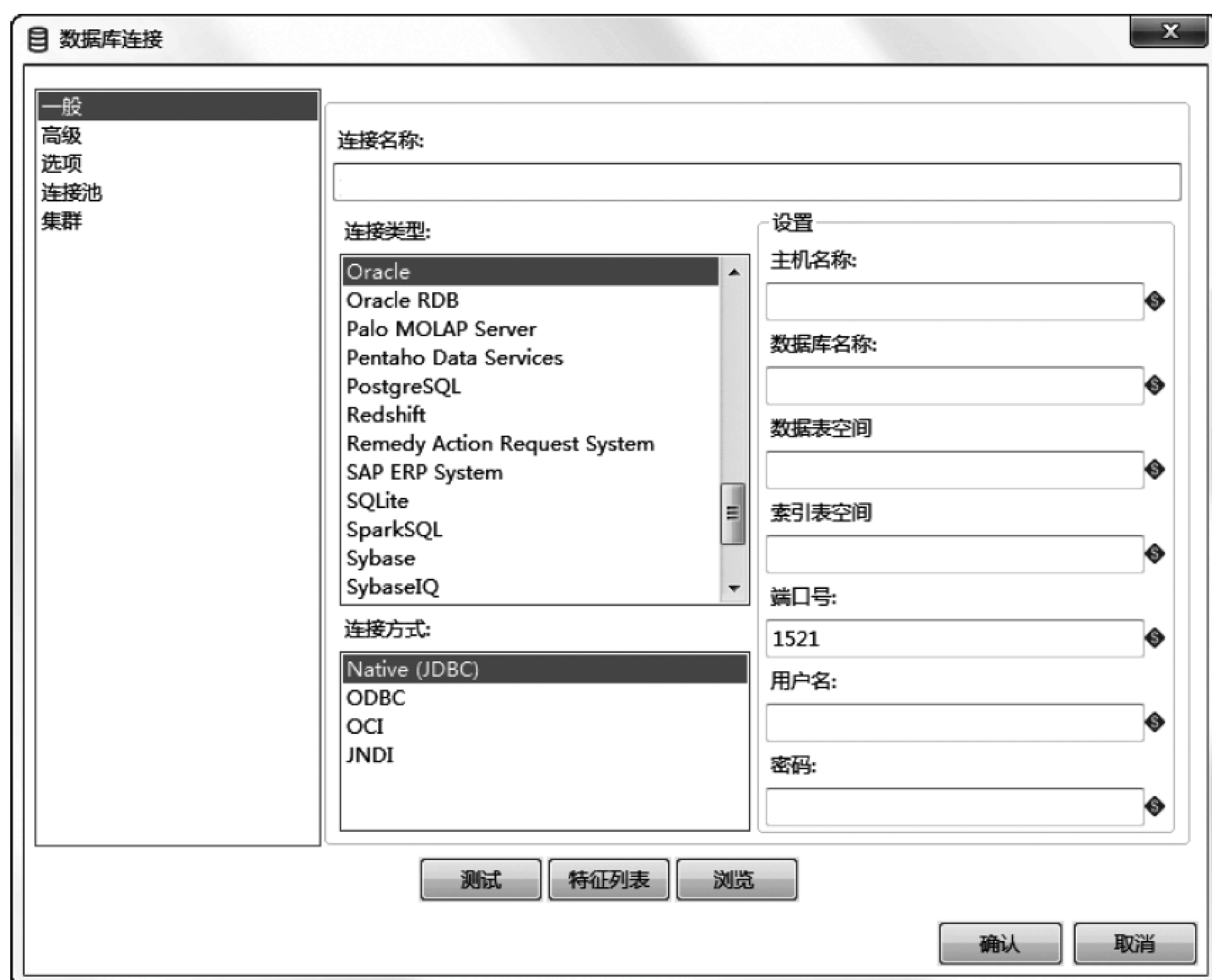


图 3-49 MySQL 数据库连接窗口

图 3-49 中，左侧面板显示的是"一般""高级""选项""连接池"以及"集群"标签，右侧面板显示的是与左侧面板标签对应的参数设置。下面针对左侧面板的标签进行详细介绍。

1. "一般"标签

单击图 3-49 中的"一般"标签，需要设置的内容有"连接名称""连接类型"以及"连接方式"等的数据库参数，具体设置规则如下。

- 连接名称：指定一个在转换或作业范围内唯一的名称。
- 连接类型：从数据库列表中选择要连接的数据库类型。根据选择数据库类型的不同，需要设置的连接方式和连接参数也会不同。Kettle 中使用某个步骤或者作业项编写 SQL 语句时，编写的 SQL 语句的语法格式也会有所不同。

• 连接方式：在连接方式列表中，可以选择与所选数据库类型对应的连接方式，一般使用 JDBC 连接，但有的数据库也使用 ODBC、JNDI、Oracle 的 OCI 连接。

由于选择的数据库不同，因此对应的图 3-49 中右侧"设置"框中需要的参数也不同。图 3-49 中显示的是连接 Oracle 数据库需要设置的连接参数。一般地，常用的数据库连接参数有主机名称、数据库名称、端口号、用户名和密码。连接参数的具体介绍如下。

• 主机名称：数据库服务器的主机名或者 IP 地址。
• 数据库名称：将要访问数据库的名称。
• 端口号：默认是选择的数据库服务器的默认端口号。
• 用户名和密码：数据库服务器的用户名和密码。

2."高级"标签

单击图 3-49 中的"高级"标签，进入"高级"标签的窗口，具体如图 3-50 所示。

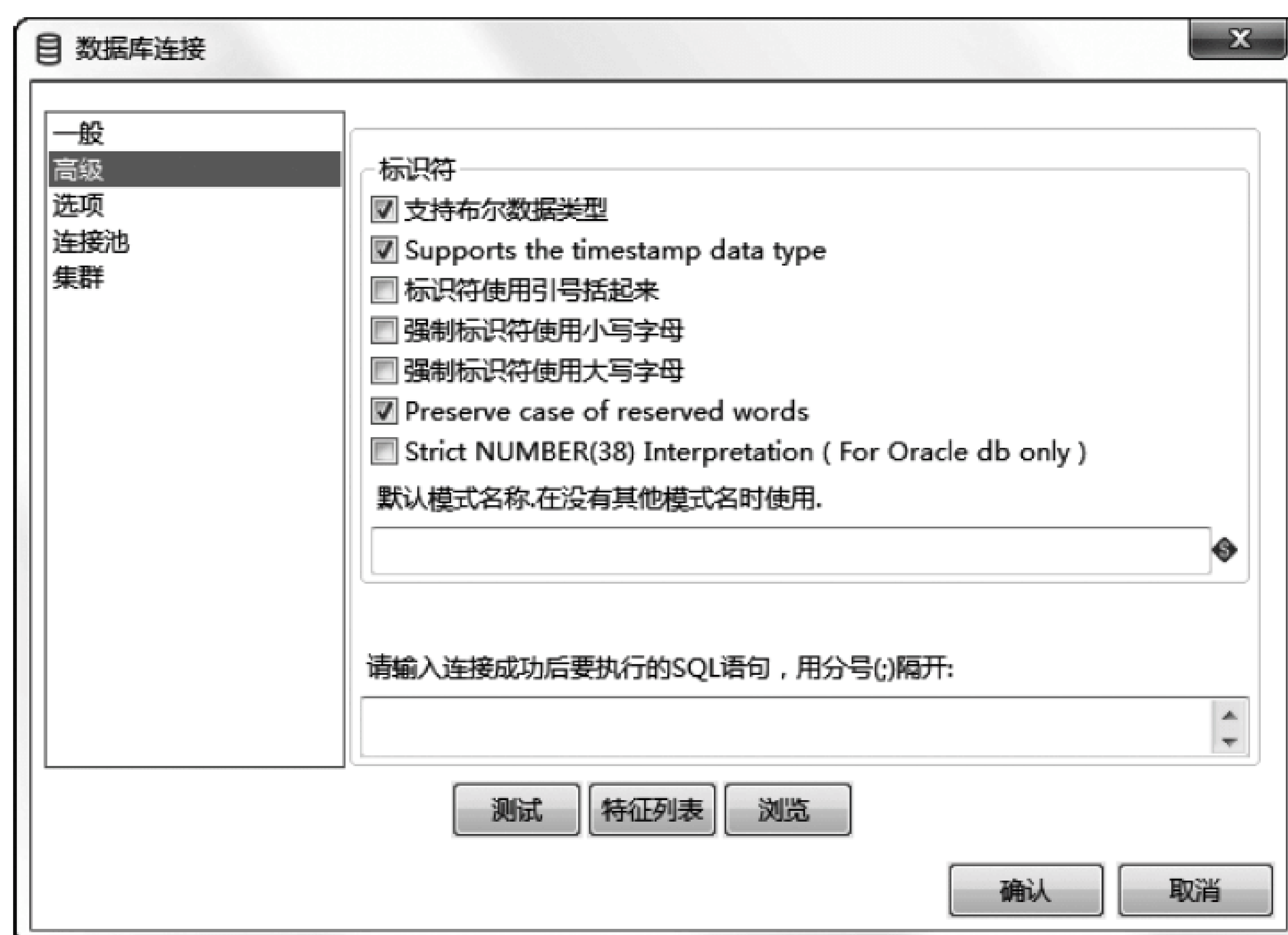


图 3-50　"高级"标签的窗口

在图 3-50 中可以设置数据库连接的标识符、默认模式的名称以及数据库连接成功后要执行的 SQL 语句，具体设置的含义如下。

(1) 支持布尔数据类型：对于 Boolean 数据类型的数据，大多数数据库的处理方式都不同，即使使用一个数据库的不同版本，也会有所不同。一般的数据库都不会支持 Boolean 类型。默认情况下，Kettle 使用一个字符的字段(即 char(1))的不同值(Y 或 N)代替 Boolean 字段。若勾选"支持布尔数据类型"复选框，Kettle 就会为支持布尔类型的数据库生成正确的 SQL 语法。

(2) Supports the timestamp data type：支持时间戳数据类型，若勾选该复选框，Kettle 就会为支持时间戳的数据库生成正确的时间类型。

(3) 标识符使用引号括起来：强制性地为 SQL 语句中的所有标识符(列名、表名)加双引号。一般地，该选项主要用于区分大小写的数据库。

(4) 强制标识符使用小写字母：将所有的标识符(列名和表名)转换为小写。

(5) 强制标识符使用大写字母：将所有的标识符(列名和表名)转换为大写。

(6) Preserve case of reserved words：保存保留字的大小写格式。

(7) Strict NUMBER(38) Interpretation：严格限制 Oracle 中 NUMBER 数据类型取值的范围(1～38)。

(8) 默认模式名称：若不明确指定模式名称(有些数据库中称为目录)，则使用默认的模式名称。

(9) 请输入连接成功后要执行的 SQL 语句：一般用于创建数据库连接后，对某些参数进行修改，如 Session 级的变量或者调试信息等。

3. “选项”标签

单击图 3-49 中的“选项”标签，进入“选项”标签的窗口，具体如图 3-51 所示。



图 3-51 “选项”标签的窗口

在图 3-51 中可以设置数据库的特定参数，如数据库连接的参数。为了便于使用，对于某些数据库(如 MySQL)，Kettle 工具提供了一些默认的连接参数和值。单击“帮助...”按钮，可以查阅对应数据库的帮助文档。

4. “连接池”标签

单击图 3-49 中的“连接池”标签，进入“连接池”标签的窗口，具体如图 3-52 所示。

在图 3-52 中可以设置连接池的相关参数，该标签的设置主要用于解决有很多小的转换(或作业)需要单独处理和数据库连接延迟的问题，连接池不会限制并发的数据库连接数量。

5. “集群”标签

单击图 3-49 中的“集群”标签，进入“集群”标签的窗口，具体如图 3-53 所示。

在图 3-53 中可以设置集群的分区数。当一个数据库不能满足需求时，可以使用多个数据库处理数据，即采用数据库分区技术分散数据的加载，这样可以将一个大的数据集分为多个小数据组(即分区)，每个分区都保存在独立的数据库中，因此，采用数据库分区技术可以减少每个数据表或数据库的行数。

注意：数据库连接只在运行转换或作业的时候使用。在作业中，每个作业项都会打开和关闭一个独立的数据库连接。转换也如此，但转换中的步骤是并行执行的，每个步骤都会

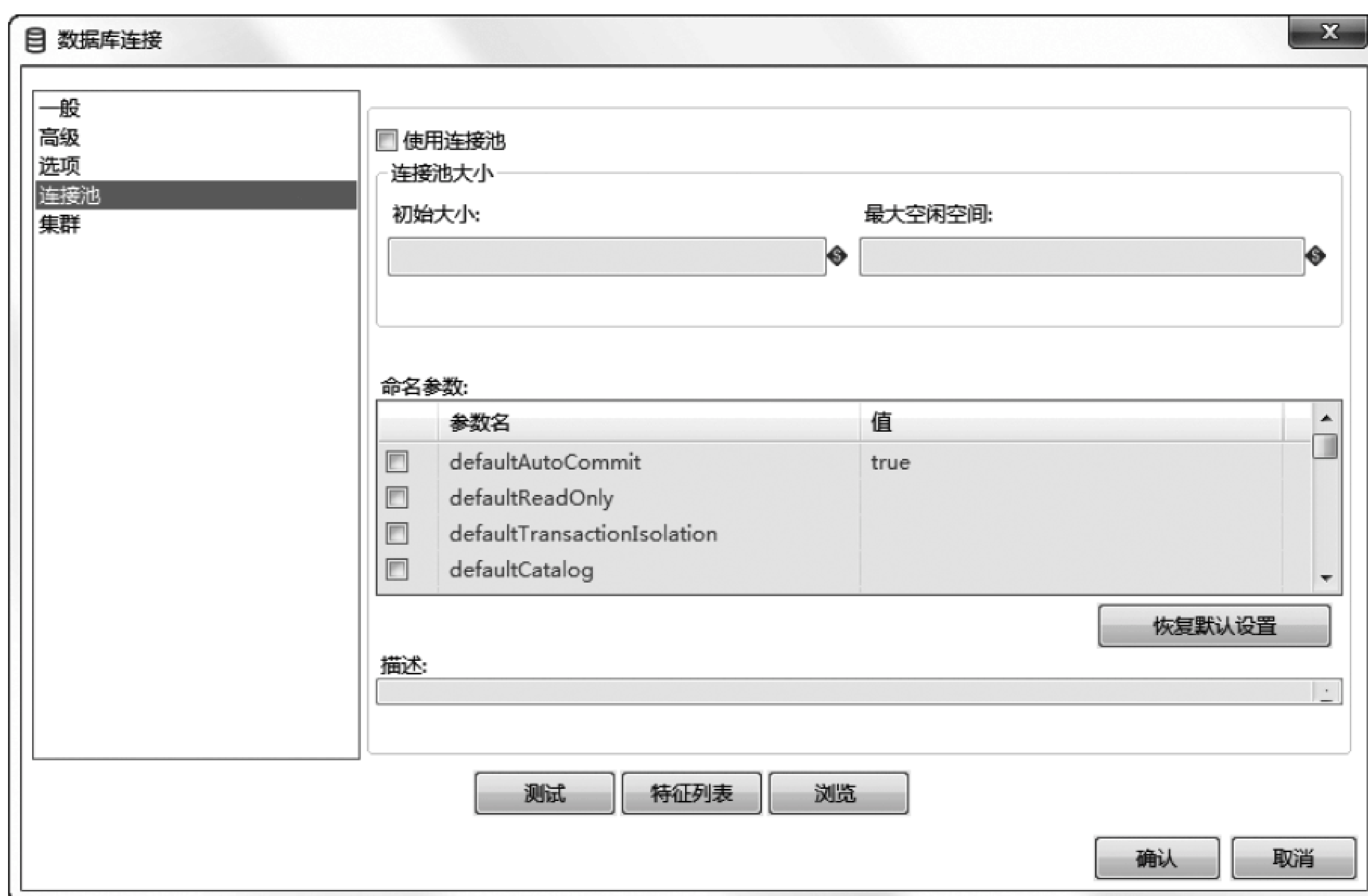


图 3-52　“连接池”标签的窗口



图 3-53　“集群”标签的窗口

打开一个独立的数据库连接，并开始一个事务。若转换中的不同步骤操作(更新或插入数据)同一张数据表，就会出现锁和参照完整性的问题。

为了解决打开多个数据库连接而产生的问题，Kettle 工具可以在一个事务中完成转换。选中转换设置对话框的选项“转换放在数据库事务中”，所有步骤里的数据库连接都使用同一个数据库连接，若所有步骤都正确，则转换正确执行，方可提交事务，否则回滚事务。

3.5　本章小结

本章主要讲解了 Kettle 工具的相关知识，包括 Kettle 简介、Kettle 的下载安装、Kettle 基本概念以及 Kettle 的基本功能。希望读者通过本章的学习，可以使用 Kettle 工具对 ETL

数据进行相关的处理。

3.6 本章习题

一、填空题

1. ________是一款国外免费开源的轻量级 ETL 工具。

2. Kettle 可以在 Windows、________、UNIX 系统上运行，并且是绿色无需安装的。

3. Kettle 的集成开发环境________提供了一个基于 SWT 的图形用户界面，主要用于 ETL 的开发。

4. 一个数据抽取过程主要包括创建一个作业，并且每个作业可以包括多个________操作。

5. 转换中的步骤是通过________连接的。

二、判断题

1. Kettle 中，数据的单位是行，数据流就是数据行从一个步骤到另一个步骤的移动。（ ）

2. Kettle 中，一个作业只包含一个作业项。（ ）

3. 转换跳是作业项之间的连接线，它定义了作业的执行路径。（ ）

4. 定义一个 Kettle 的数据库连接，是要真正打开一个到数据库的连接。（ ）

5. 作业执行的顺序由作业项之间的跳和每个作业项的执行结果决定。（ ）

三、选择题

1. 下列设计原则，选项________不属于 Kettle 的设计原则。

   A. 避免自定义开发　　B. 灵活的数据通道

   C. 可维护性与可扩展性的原则　　D. 只映射需要映射的字段

2. 下列类别，________不属于作业管理。

   A. 邮件　　B. 文件管理　　C. 条件　　D. 应用

3. 下列组件中，________属于 Kettle 工具。

   A. Spoon　　B. Pan　　C. Kitchen　　D. Carte

四、操作题

通过 Kettle 工具，实现以下功能：

（1）将一张数据表的两个字段进行拼接，然后插入另一张数据表中。

（2）发送邮件。

第4章 数据抽取

学习目标

（1）掌握抽取文本数据的方法
（2）掌握抽取 Web 数据的方法
（3）掌握抽取数据库数据的方法

在数据清洗中，数据抽取主要是从一个或多个数据源中获取所需的数据。数据抽取的数据源可以是文本数据、Web 数据以及数据库数据。本章将针对文本数据、Web 数据以及数据库数据的抽取操作进行详细讲解。

4.1 抽取文本数据

在实际应用中，常用的文本文件类型有两种，分别是 TSV 文件和 CSV 文件。本节将针对抽取 TSV 类型和 CSV 类型的文本文件数据分别进行讲解。

4.1.1 TSV 文件的抽取

TSV 是 Tab-Separated Values 的缩写，即制表符分隔值。使用制表符分隔数据字段的文件被称为制表符文件。制表符文件中的数据以表格结构存储，每行存储一条记录，每条记录的各个字段间使用制表符分隔。大多数的操作系统和常用文本编辑器中，按一次 Tab 键表示默认插入一个制表符。

一般情况下，制表符的种类包含左对齐式制表符、居中式制表符、右对齐式制表符、小数点对齐式制表符和竖线对齐式制表符等种类。通过设置不同制表符的位置，在输入一项数据之后，按一下 Tab 键，光标会根据制表符的设置在数据后面插入制表符。通过制表符分隔的文本数据与未使用制表符分隔的数据相比，前者更便于观察识别，同时也便于对数据进行抽取操作。

现有一个 TSV 文件 tsv_extract.tsv，具体内容如图 4-1 所示。

```
1    itcast    2019-04-28
2    hadoop    2019-04-30
3    heima     2019-05-28
4    spark     2019-05-29
5    kettle    2019-05-30
```

图 4-1 TSV 文件 tsv_extract.tsv

下面分步骤讲解如何抽取 TSV 文件 tsv_extract.tsv 中的数据并保存至数据库 extract 中的数据表 tsv 中，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建一个转换 tsv_extract，添加“文本文件输入”控件、“表输出”控件以及 Hop 跳连接线，用于实现 TSV 文件数据的抽取功能，具体效果如图 4-2 所示。

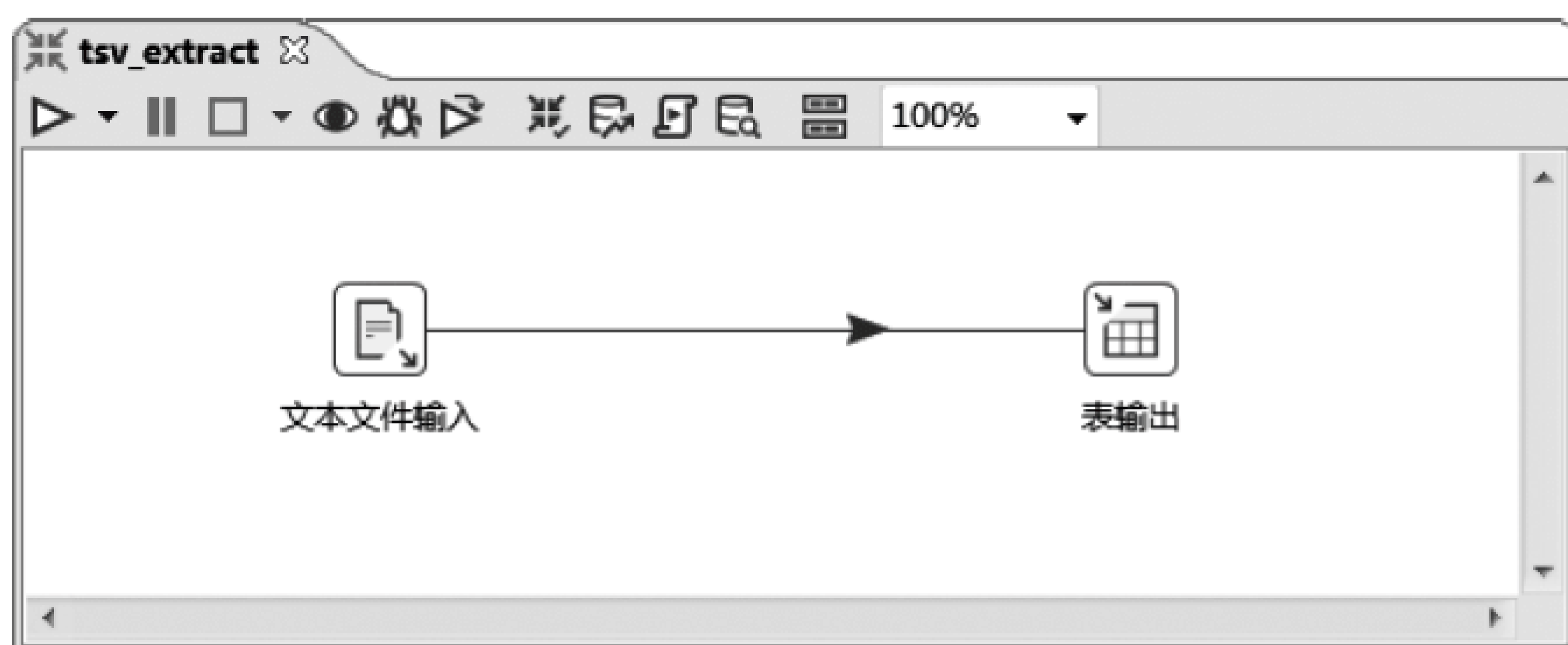


图 4-2 创建转换 tsv_extract

2. 配置“文本文件输入”控件

双击图 4-2 中的“文本文件输入”控件，进入“文本文件输入”界面，具体如图 4-3 所示。



图 4-3 “文本文件输入”界面

单击图 4-3 中的“浏览”按钮，选择要抽取的文件 tsv_extract.tsv，效果如图 4-4 所示。

单击图 4-4 中的“增加”按钮，将要抽取的 TSV 文件添加到转换 tsv_extract 中，具体效果如图 4-5 所示。

单击图 4-5 中的“内容”选项卡，切换到“内容”选项卡界面，如图 4-6 所示。

在图 4-6 中清除分隔符处的默认分隔符“;”，并单击 Insert TAB 按钮，在分隔符处插入一个制表符；取消勾选“头部”复选框，若不取消，在进行数据抽取操作时会排除文件第一行的数据。“内容”选项卡的配置如图 4-7 所示。

单击图 4-7 中的“字段”选项卡，切换到“字段”选项卡界面，如图 4-8 所示。



图 4-4　选择要抽取的文件 tsv_extract. tsv

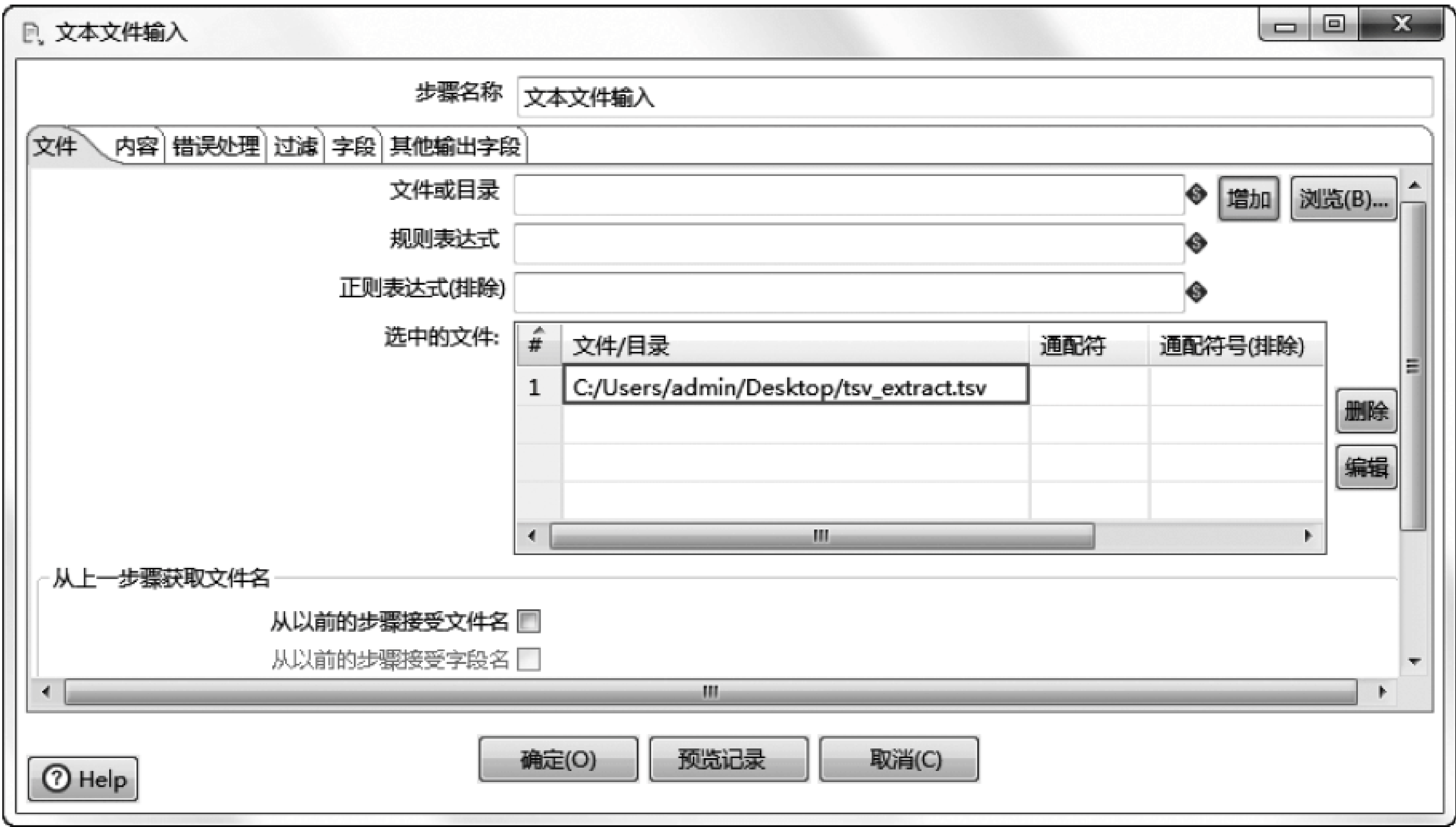


图 4-5　添加 TSV 文件至转换 tsv_extract 中

在图 4-8 中，根据 TSV 文件的内容添加对应的字段名称，并指定数据类型。这里需要注意的是，制表符可看作由多个空格组成，因此，在“去除空字符串方式”列时，所添加的字段都应选择“不去除空格”，否则进行抽取数据操作时会把制表符当作空格去除，而不能把制表符作为分隔符实现文本文件内容的分隔。“字段”选项卡的配置如图 4-9 所示。

单击图 4-9 中的“预览记录”按钮，查看文件 tsv_extract. tsv 的数据是否成功抽取到文本文件输入流中，具体效果如图 4-10 所示。

从图 4-10 中可以看出，文件 tsv_extract. tsv 的数据已经成功抽取到文本文件输入流中，单击“关闭”→“确定”按钮，完成“文本文件输入”控件的配置。



图 4-6 “内容”选项卡界面



图 4-7 “内容”选项卡的配置



图 4-8 “字段”选项卡界面



图 4-9　“字段”选项卡的配置

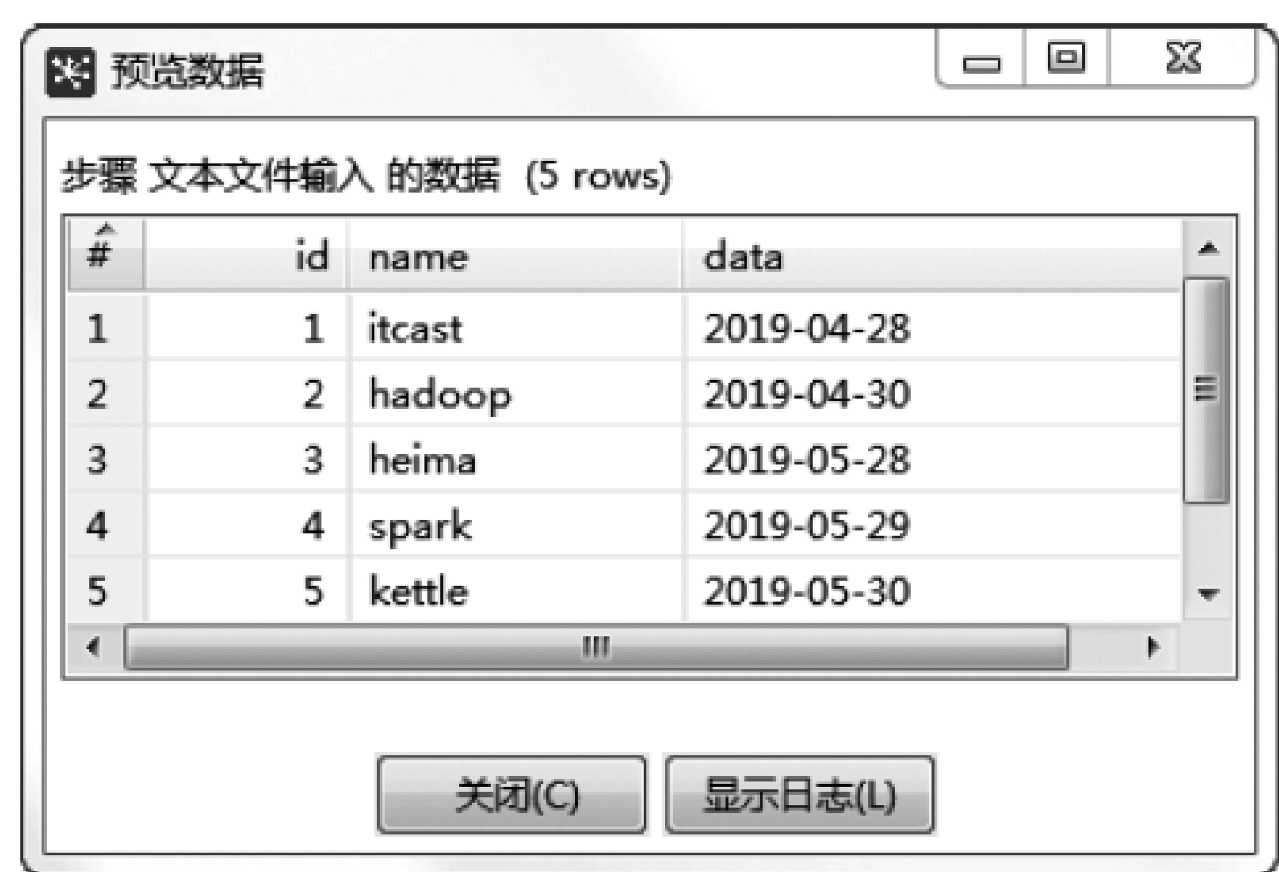


图 4-10　预览数据

3. 配置“表输出”控件

双击图 4-2 中的“表输出”控件，进入“表输出”界面，具体如图 4-11 所示。



图 4-11　“表输出”界面

单击图 4-11 中的“新建”按钮，配置数据库连接(连接的数据库 extract 需提前创建，这里不再赘述)，配置完成后单击“确认”按钮。MySQL 数据库连接的配置具体如图 4-12 所示。



图 4-12　MySQL 数据库连接的配置

单击图 4-11 中目标表右侧的“浏览”按钮，指定输出目标表，即数据表 tsv（该表需提前创建，且表结构需根据文件 tsv_extract. tsv 中的字段、类型进行创建，这里不演示），如图 4-13 所示。



图 4-13　指定输出的目标表

勾选图 4-13 中的“指定数据库字段”复选框，用于将数据表 tsv 的字段与文件 tsv_extract. tsv 中的字段进行匹配，具体如图 4-14 所示。

单击图 4-14 中的“数据库字段”选项卡，具体如图 4-15 所示。

单击图 4-15 中的“输入字段映射”按钮，弹出“映射匹配”对话框，具体如图 4-16 所示。



图 4-14　勾选“指定数据库字段”复选框

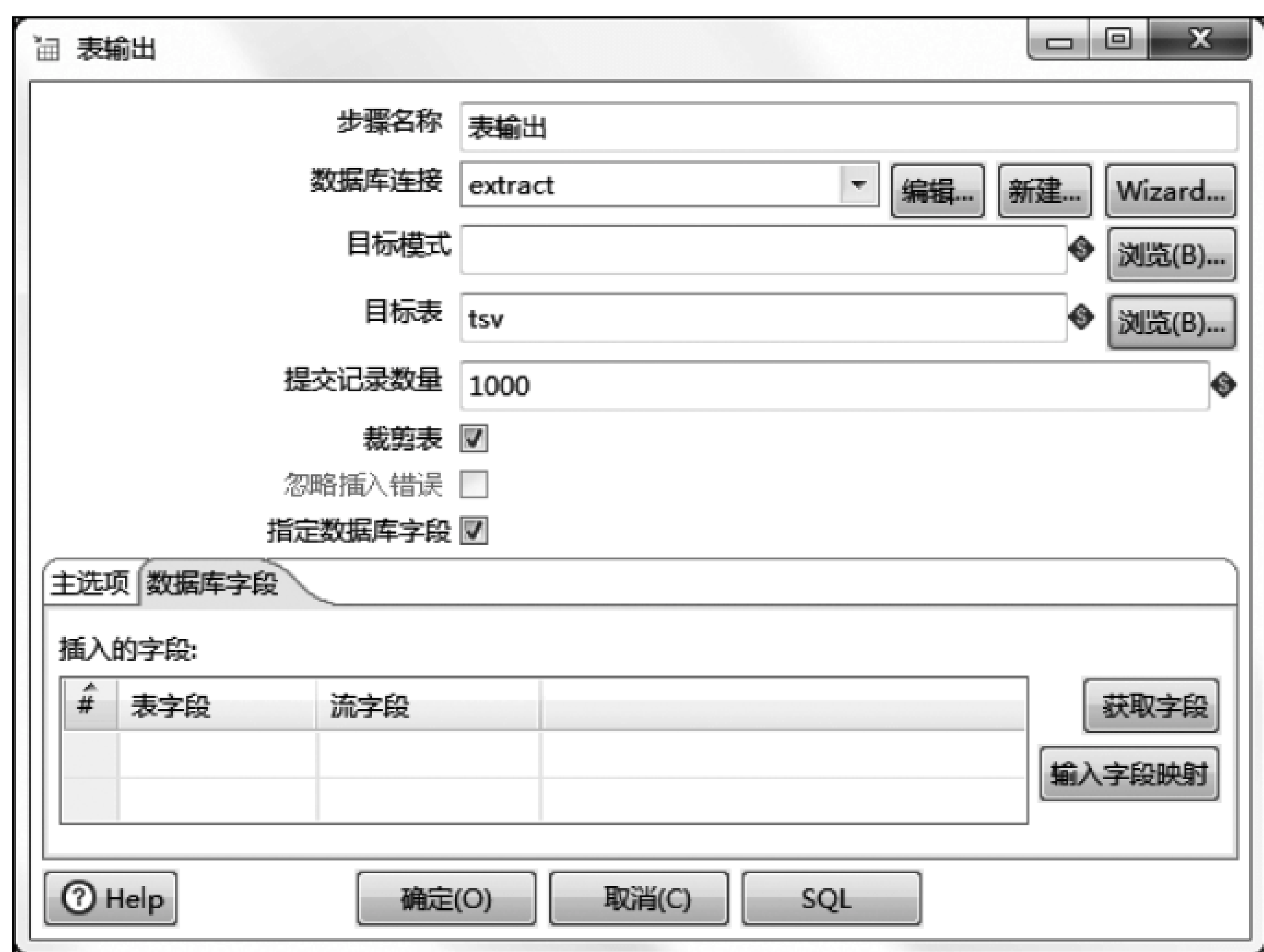


图 4-15　“数据库字段”选项卡

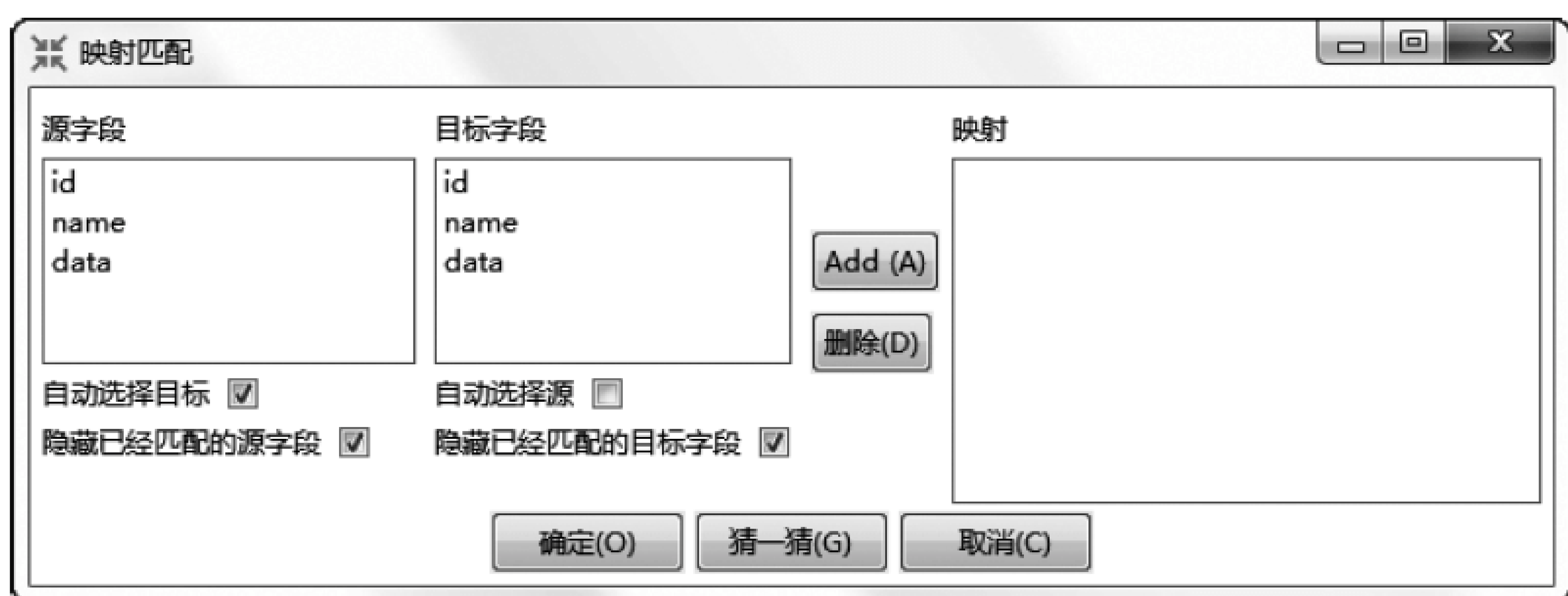


图 4-16　“映射匹配”对话框

在图 4-16 中依次选中“源字段”中的字段和“目标字段”中对应的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 4-17 所示。



图 4-17 设置映射匹配

单击图 4-17 中的“确定”按钮，“表输出”控件配置的效果图如图 4-18 所示。



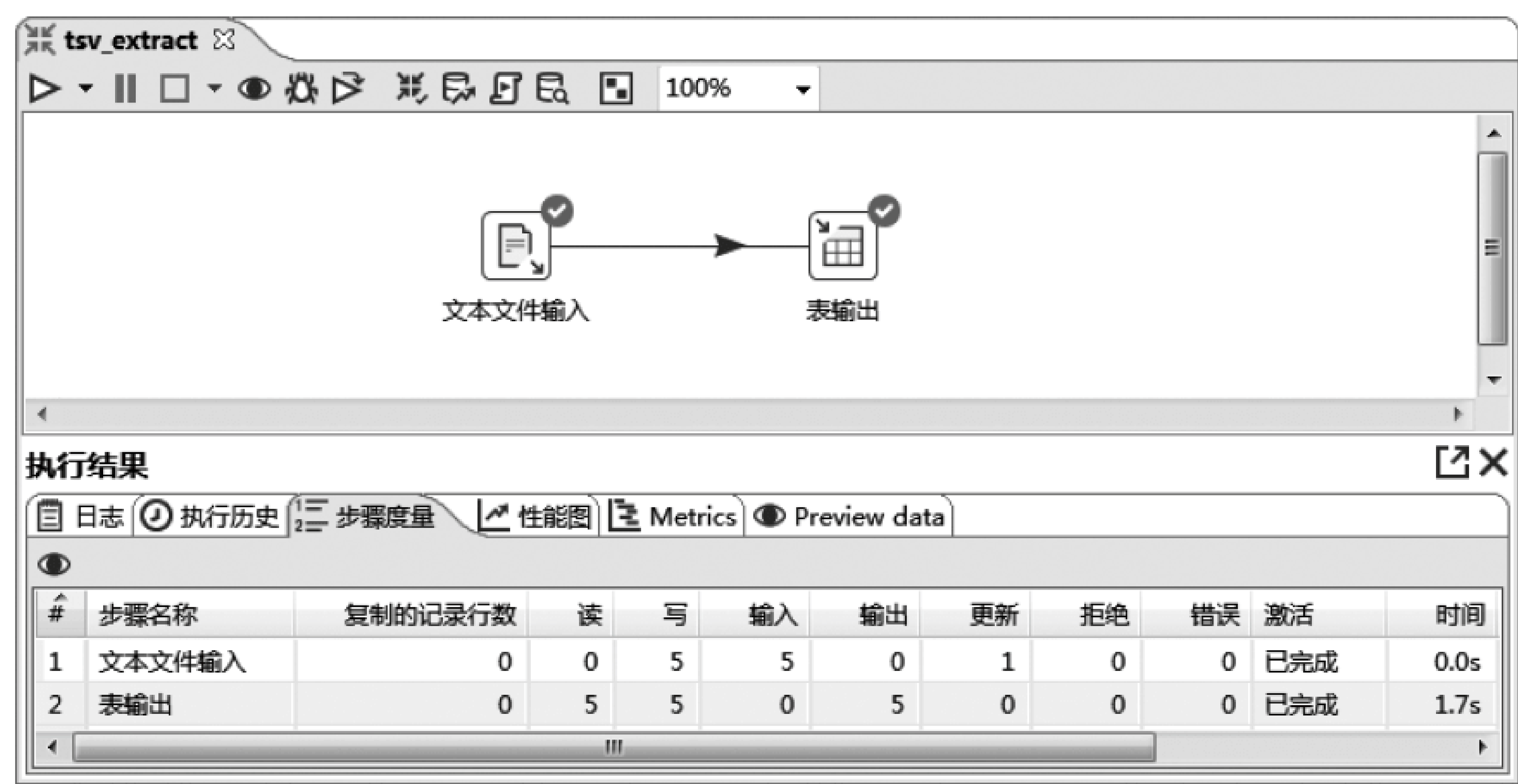
图 4-18 “表输出”控件配置的效果图

单击图 4-18 中的“确定”按钮，完成“表输出”控件的配置。

4. 运行转换 tsv_extract

单击转换工作区顶部的 ▷ 按钮，运行创建的转换 tsv_extract，实现将 TSV 文件中的数据抽取到数据表 tsv 中，具体如图 4-19 所示。

从图 4-19 中执行结果的“步骤度量”可以看出，“文本文件输入”控件输入 5 条数据并写入该控件中；而“表输出”控件读取“文本文件输入”控件中的 5 条数据并写入该控件中，最终进行输出。也就是说，“表输出”控件将从文本文件输入流中读取的 5 条数据均写入数据表 tsv 中。

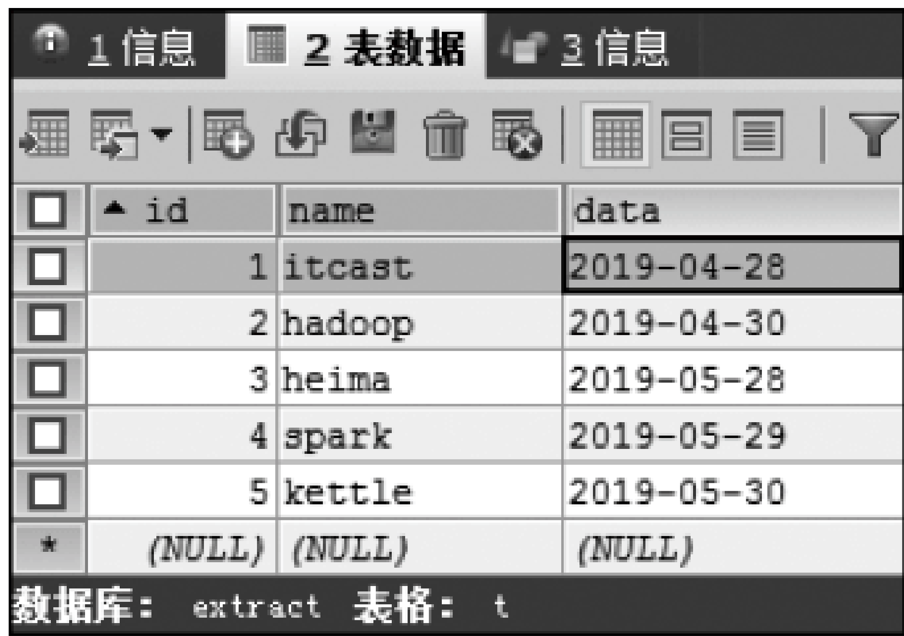


#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活	时间
1	文本文件输入	0	0	5	5	0	1	0	0	已完成	0.0s
2	表输出	0	5	5	0	5	0	0	0	已完成	1.7s

图 4-19　运行转换 tsv_extract

5. 查看 tsv 数据表中的数据

通过 SQLyog 工具，查看 tsv 数据表是否已成功插入 5 行数据，查看结果如图 4-20 所示。



id	name	data
1	itcast	2019-04-28
2	hadoop	2019-04-30
3	heima	2019-05-28
4	spark	2019-05-29
5	kettle	2019-05-30
(NULL)	(NULL)	(NULL)

数据库: extract 表格: t

图 4-20　tsv 数据表

从图 4-20 中可以看出，tsv 数据表中已插入 5 行数据，说明成功实现了将 TSV 文件 tsv_extract.tsv 中的数据抽取到 tsv 数据表中。

4.1.2　CSV 文件的抽取

CSV 是 Comma-Separated Values 的缩写，即逗号分隔值。CSV 文件是用逗号分隔数据字段的文件，因此也被称为逗号分隔值文件，有时会使用字符替代逗号实现分隔，因此也被称为字符分隔文件。CSV 文件是以纯文本形式存储表格数据（数字和文本），纯文本意味着该文件是一个字符序列。CSV 文件可通过 Excel 打开，也可通过 txt、Notepad++ 等文本编辑器打开，从而对文件进行查看、编辑等操作。

CSV 文件由任意数目的记录组成，记录之间以某种换行符分隔；每条记录由字段组成，字段之间的分隔符常见的有逗号或制表符。通常，整个文件中的所有记录都有完全相同的字段序列。

CSV 作为数据转存的一种常用格式，有特定的实现规则，具体包含以下 8 点：

- 文件开头不能留空，以“行”为单位。
- 文件可含或不含列名，若含有列名，则位于文件第一行。
- 文件中的一行数据不能跨行，行与行间不可存在空行。
- 文件中以英文半角逗号(即“,”)作为分隔符，若列为空，也要表达空列的存在。
- 文件中的列内容，若存在英文半角单引号(即“'”)，则替换成半角双引号(即“""”)进行转义，因为在抽取数据时，通过使用半角双引号“"”将所有的字符串内容引起来。
- 在文件读写时，引号和逗号操作规则可以互逆。

- 文件中的编码格式不做限制，可以是 ASCII，也可以是 Unicode 或者 UTF8 等编码格式。
- 文件中不支持数字或特殊字符。

如果想实现抽取 CSV 文件中的数据，必须遵循以上 8 项规则。在抽取数据的过程中，通过使用默认提供的逗号分隔符或其他分隔符，抽取每条记录中的字段值。

现有一个 CSV 文件 csv_extract. csv，使用 Excel 打开的效果如图 4-21 所示。

	A	B	C	D	E
1	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
2	1	Male	19	15	39
3	2	Male	21	15	81
4	3	Female	20	16	6
5	4	Female	23	16	77
6	5	Female	31	17	40
7	6	Female	22	17	76
8	7	Female	35	18	6
9	8	Female	23	18	94
10	9	Male	64	19	3
11	10	Female	30	19	72
12	11	Male	67	19	14
13	12	Female	35	19	99
14	13	Female	58	20	15
15	14	Female	24	20	77
16	15	Male	37	20	13
17	16	Male	22	20	79
18	17	Female	35	21	35
19	18	Male	20	21	66
20	19	Male	52	23	29
21	20	Female	35	23	98

csv_extract

图 4-21 csv_extract. csv 文件的部分数据

下面分步骤讲解如何抽取 CSV 文件 csv_extract. csv 中的数据并保存至数据库 extract 的数据表 csv 中，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建一个转换 csv_extract，并添加“CSV 文件输入”控件、“表输出”控件以及 Hop 跳连接线，用于实现 CSV 文件数据的抽取功能，具体效果如图 4-22 所示。

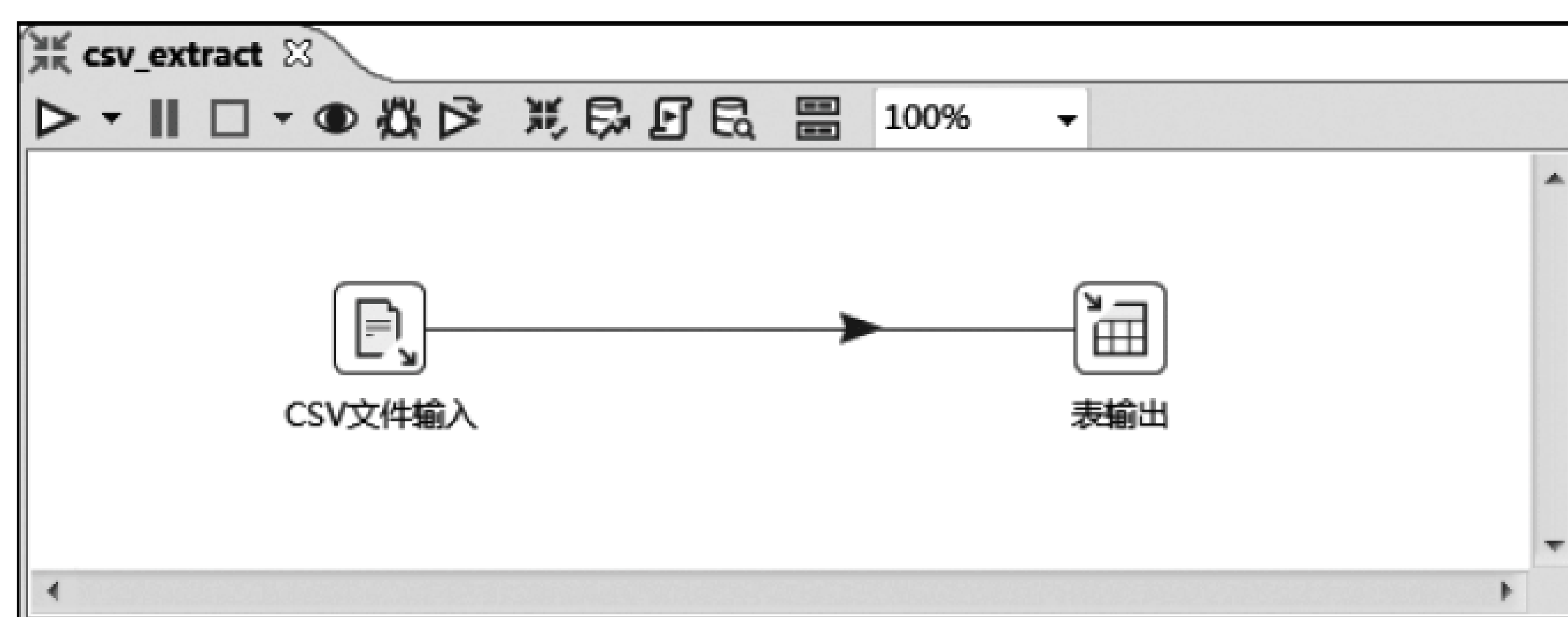


图 4-22 创建转换 csv_extract

2. 配置“CSV 文件输入”控件

双击图 4-22 中的“CSV 文件输入”控件，进入“CSV 文件输入”界面，具体如图 4-23

所示。



图 4-23　“CSV 文件输入”界面

单击图 4-23 中的“浏览”按钮，选择要抽取的文件 csv_extract. csv，具体如图 4-24 所示。



图 4-24　选择要抽取的文件 csv_extract. csv

单击图 4-24 中的“获取字段”按钮，Kettle 自动检索 CSV 文件，并对文件中的字段类型、格式、长度、精度等属性进行分析，具体效果如图 4-25 所示。

单击图 4-25 中的“预览”按钮，查看文件 csv_extract. csv 的数据是否抽取到 CSV 文件输入流中，具体效果如图 4-26 所示。

从图 4-26 中可以看出，CSV 文件 csv_extract. csv 的数据已经成功抽取到 CSV 文件输入流中，单击“关闭”→“确定”按钮，完成“CSV 文件输入”控件的配置。



图 4-25 获取字段

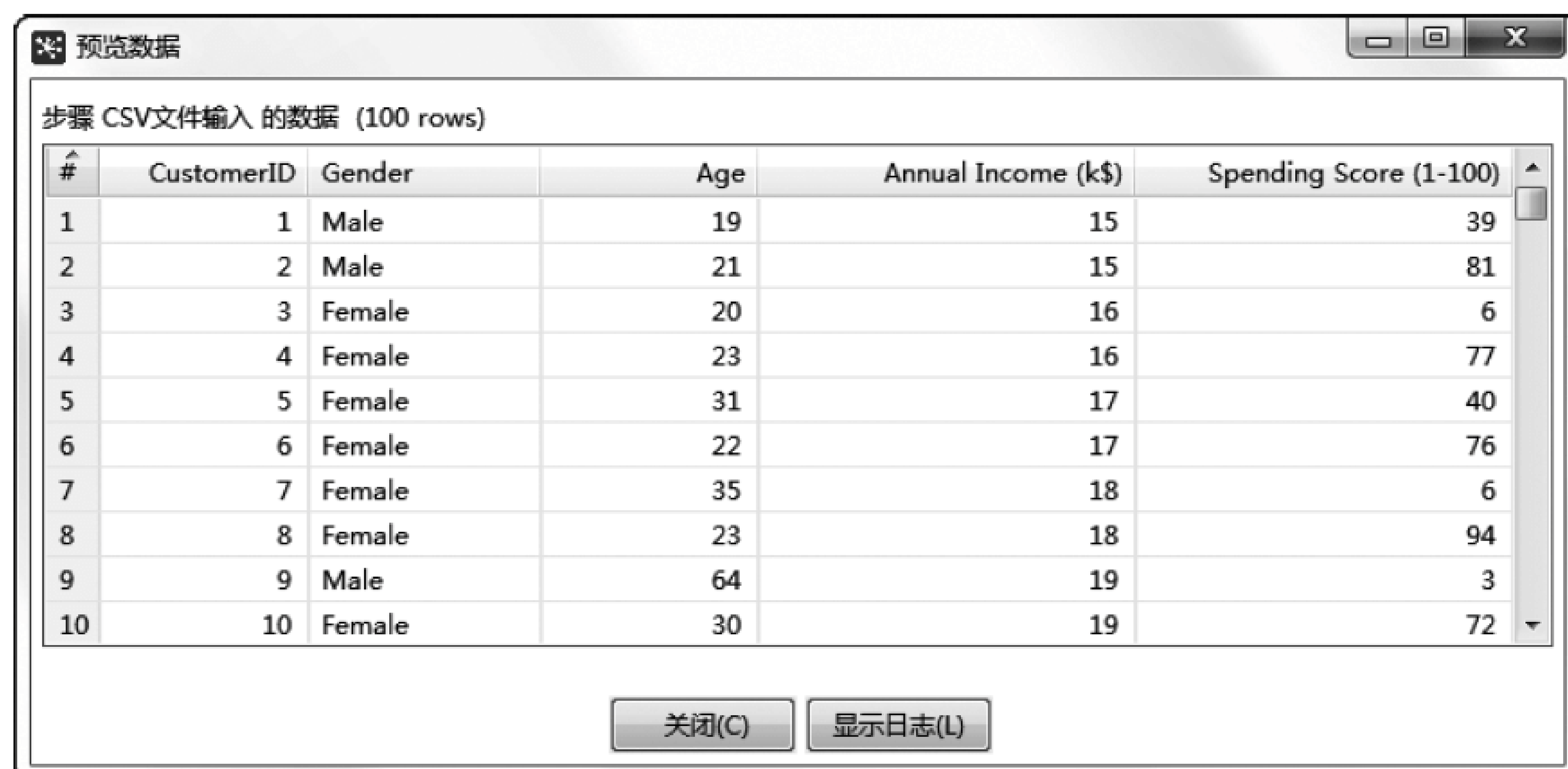


图 4-26 预览数据

3. 配置“表输出”控件

双击图 4-22 中的“表输出”控件，进入“表输出”界面，具体如图 4-27 所示。

单击图 4-27 中的“新建”按钮，配置数据库连接(所连接的数据库 extract 需提前创建，这里不再赘述)，配置完成后单击“确认”按钮。MySQL 数据库连接的配置具体如图 4-28 所示。

单击图 4-27 中目标表右侧的“浏览”按钮，选择输出的目标表，即数据表 csv_extract(该表需提前创建，且表结构需根据 CSV 文件 csv_extract. csv 中数据的字段和数据类型进行创建，这里不演示)；勾选“指定数据库字段”复选框，用于将数据表 csv 的字段与 CSV 文件



图 4-27　“表输出”界面



图 4-28　MySQL 数据库连接的配置

csv_extract.csv 中的字段进行匹配,具体如图 4-29 所示。

在图 4-29 中选择“数据库字段”选项卡,具体如图 4-30 所示。

单击图 4-30 中的“输入字段映射”按钮,弹出“映射匹配”对话框,具体如图 4-31 所示。

在图 4-31 中依次选中“源字段”中的字段和“目标字段”中的对应字段,再单击 Add(A) 按钮,将一对映射字段添加至“映射”框中,若“源字段”中的字段和“目标字段”中的字段相同,则可以单击“猜一猜”按钮,让 Kettle 自动实现映射,具体如图 4-32 所示。

单击图 4-32 中的“确定”按钮,“表输出”控件的配置效果如图 4-33 所示。

单击图 4-33 中的“确定”按钮,完成“表输出”控件的配置。



图 4-29 指定输出的目标表



图 4-30 “数据库字段”选项卡

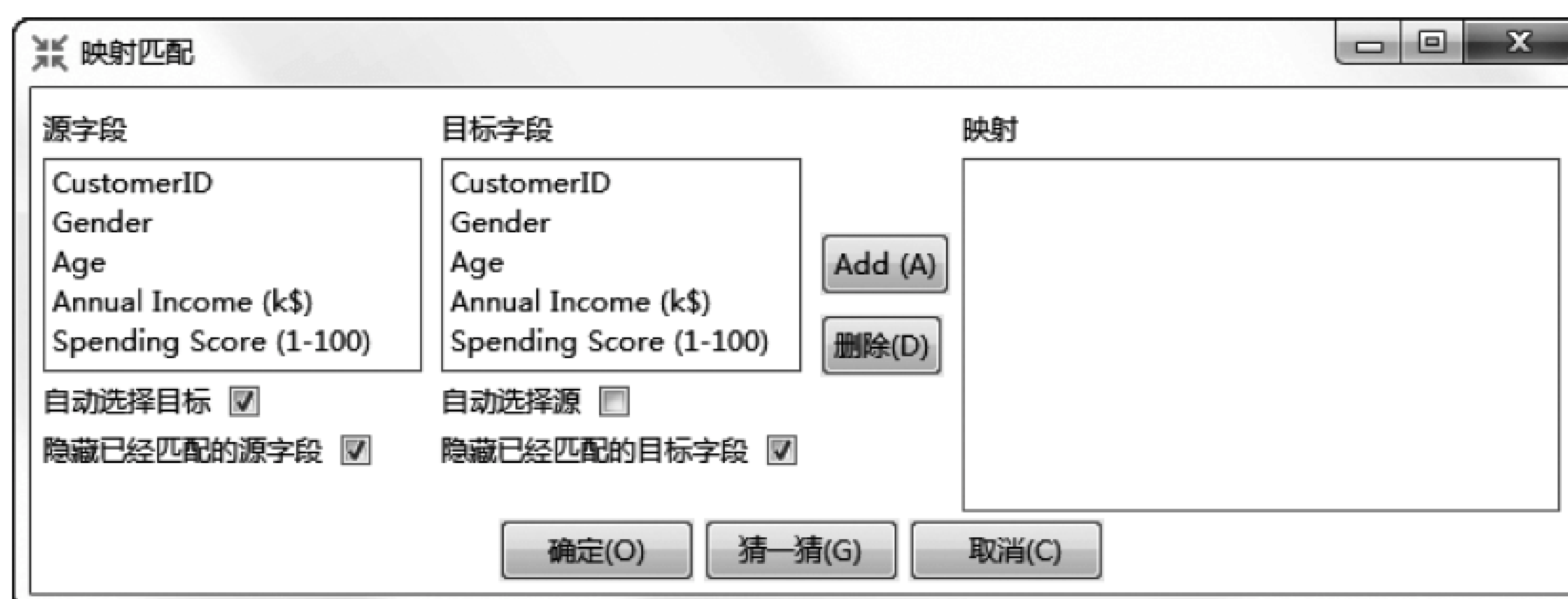


图 4-31 “映射匹配”对话框



图 4-32　设置映射匹配



图 4-33　“表输出”控件的配置效果

4. 运行转换 csv_extract

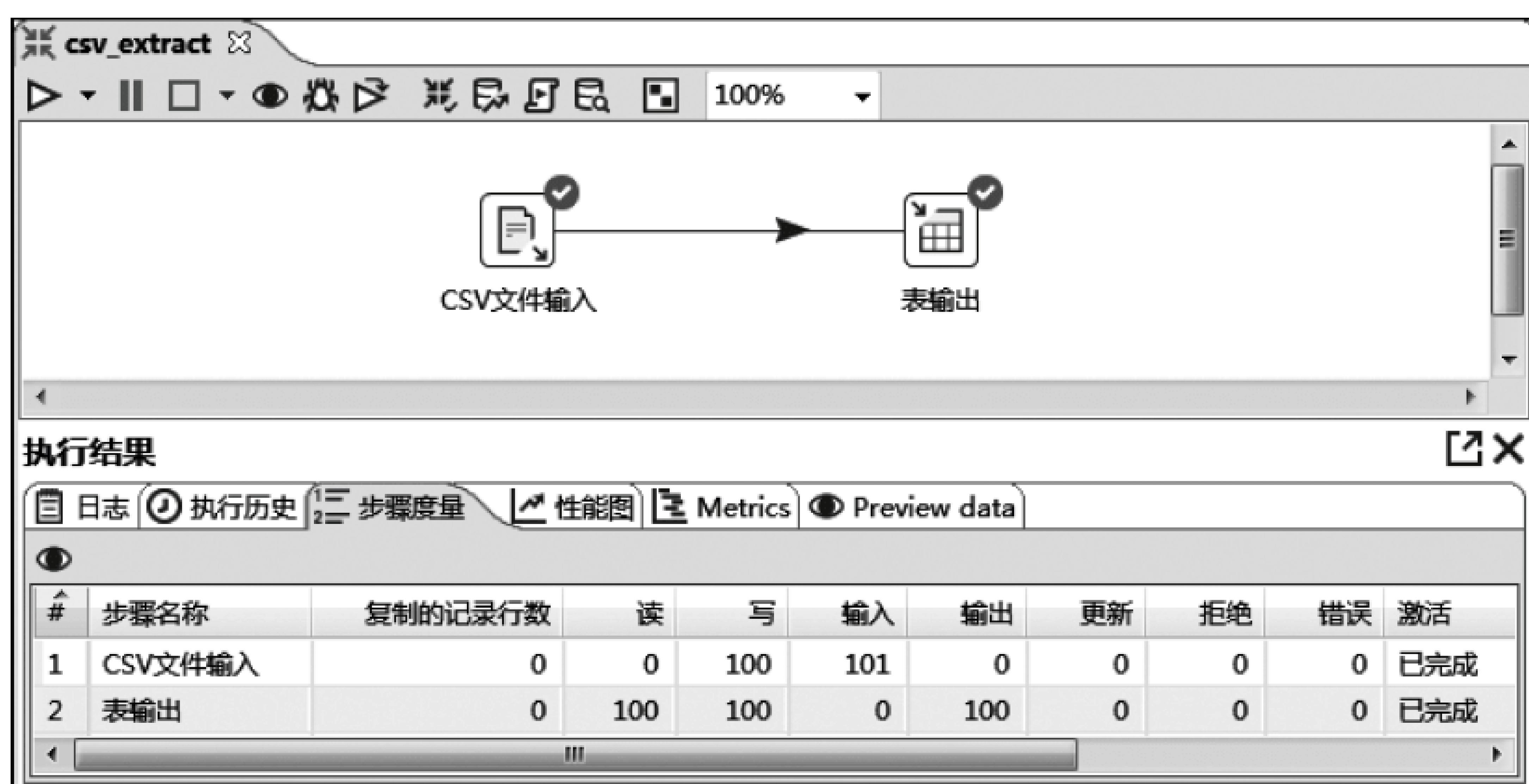
单击转换工作区顶部的▷按钮，运行创建的转换 csv_extract，实现将 CSV 文件中的数据抽取到数据表 csv 中，具体如图 4-34 所示。

从图 4-34 中执行结果的“步骤度量”可以看出，“CSV 文件输入”控件输入 101 条数据并写入该控件 100 条数据(其中有 1 条数据为表头数据)，而“表输出”控件读取“CSV 文件输入”控件中的 100 条数据并写入该控件，最终进行输出。也就是说，“表输出”控件将从 CSV 文件输入流中读取的 100 条数据均写入数据表 csv 中。

5. 查看数据表 csv 中的数据

通过 SQLyog 工具，查看数据表 csv 是否已成功插入 100 行数据，结果如图 4-35 所示。

从图 4-35 中可以看出，数据表 csv 中已插入数据(这里只展示数据表中的部分数据)，说明我们成功实现了将 CSV 文件 csv_extract. csv 中的数据抽取到数据表 csv 中。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	CSV文件输入	0	0	100	101	0	0	0	0	已完成
2	表输出	0	100	100	0	100	0	0	0	已完成

图 4-34　运行转换 csv_extract

CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72
11	Male	67	19	14
12	Female	35	19	99
13	Female	58	20	15
14	Female	24	20	77
15	Male	37	20	13

数据库: extract 表格: csv

图 4-35　数据表 csv

4.2　抽取 Web 数据

抽取 Web 数据主要是获取网页上的数据。Web 网页上出现的数据形式主要有 3 种，分别是 HTML 形式、XML 形式以及 JSON 形式。本节将针对抽取 HTML、XML 以及 JSON 三种形式文件中的数据分别进行详细讲解。

4.2.1　HTML 网页的数据抽取

HyperText Markup Language，简称 HTML，即超文本标记语言，它包含了一套标记标签，主要用于创建和描述网页。HTML 可以以文档的形式展示。HTML 文档中包含 HTML 标签和纯文本。其中，HTML 标签是由尖括号括起来的关键词，如<html>和</html>、<head>和</head>、<body>和</body>等，这些标签通常以第一个标签（如<html>标签）为开始标签，第二个标签（如</html>标签）为结束标签的方式成对出

现。在标签内部可以定义 id,用于标签的唯一标识;也可以定义 class,用于一组标签的标识。

在数据抽取过程中,由于 HTML 网页本身是无结构化或半结构化的,因此 HTML 网页中的数据面临着较大的抽取困难。基于数据库技术的 HTML 网页抽取技术的研究经过了人工、半自动化和全自动化方法的 3 个阶段。

- 人工方法,通过程序员人工分析出网页的模板,借助一定的编程语言,针对具体的问题生成具体的包装器。
- 半自动化方法,应用网页模板抽取数据,从而生成具体包装器的部分被计算机接管,而网页模板的分析仍然需要人工参与。
- 全自动化方法中,网页模板的分析部分也交给了计算机进行,仅需要很少的人工参与或完全不需要人工参与,因而更加适合大规模、系统化、持续性的 Web 数据抽取。

下面通过人工方法实现 HTML 网页数据的抽取,即人工对网页源码的结果分析,借助编程语言,使用正则表达式匹配 HTML 中的标签和标签属性,将有价值的数据抽取出来。

这里以抽取“豆瓣电影排行榜”网页的超链接数据为例,分步骤讲解如何抽取 HTML 网页的数据,并保存至数据库 extract 中的数据表 html(该表需要提前创建,这里不再赘述)中。豆瓣电影排行榜页面的部分内容如图 4-36 所示。



图 4-36 豆瓣电影排行榜页面的部分内容

实现抽取 HTML 网页的数据,并保存至数据库 extract 的数据表 html 中,具体步骤如下。

1. 打开 Kettle 工具,创建转换

使用 Kettle 工具创建一个转换 html_extract,并添加“自定义常量数据”输入控件、“HTTP client”查询控件和“Java 代码”脚本控件,用于实现抽取 HTML 网页的 Web 数据的功能,具体效果如图 4-37 所示。

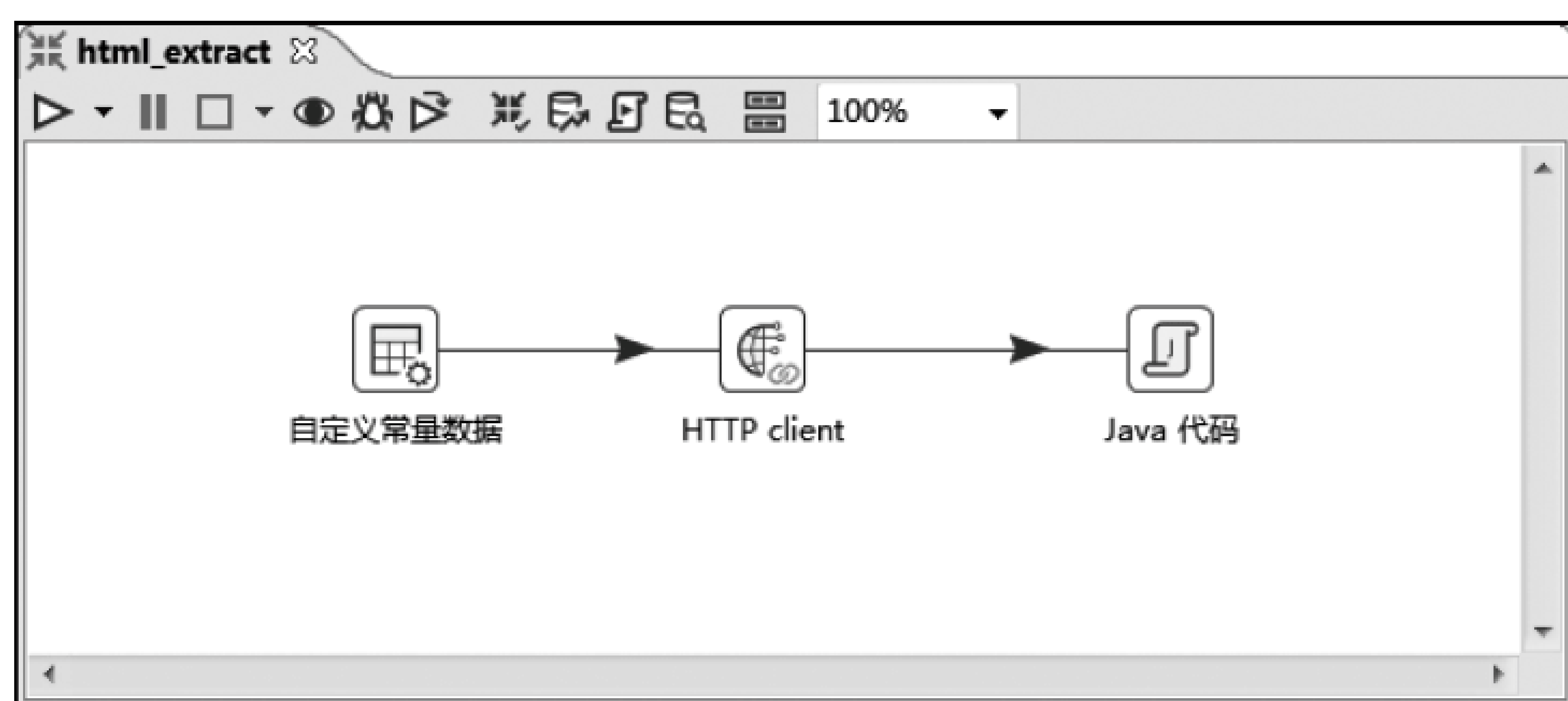


图 4-37 创建转换 html_extract

2. 配置“自定义常量数据”控件

双击图 4-37 中的“自定义常量数据”控件，进入“自定义常量数据”界面，具体如图 4-38 所示。



图 4-38 “自定义常量数据”界面

单击图 4-38 中的“元数据”选项卡，定义一个字段常量 filename 并指定数据类型 String；单击“数据”选项卡，添加 html 形式数据所在的 URL，即 https://movie.douban.com/chart，具体效果如图 4-39 所示。



图 4-39 “自定义常量数据”控件配置的效果图

单击图 4-39 中的“确定”按钮，完成“自定义常量数据”控件的配置。

3. 配置 HTTP client 控件

双击图 4-37 中的 HTTP client 控件，进入 HTTP web service 界面，具体如图 4-40 所示。



图 4-40　HTTP web service 界面

勾选图 4-40 中的“从字段中获取 URL?”复选框；在“URL 字段名”后的下拉列表中选择 URL 字段名，即 filename；在“结果字段名”处指定结果字段名称，这里选择默认的结果字段 result。HTTP client 控件配置的效果，具体如图 4-41 所示。



图 4-41　HTTP client 控件配置的效果图

单击图 4-41 中的“确定”按钮，完成 HTTP client 控件的配置。

4．配置“Java 代码”控件

双击图 4-37 中的“Java 代码”控件，进入“Java 代码”界面，具体如图 4-42 所示。



图 4-42 “Java 代码”界面

双击图 4-42 中的 Code Snippits→Common use→Main，添加 Java 脚本代码的主方法，即程序入口，具体效果如图 4-43 所示。

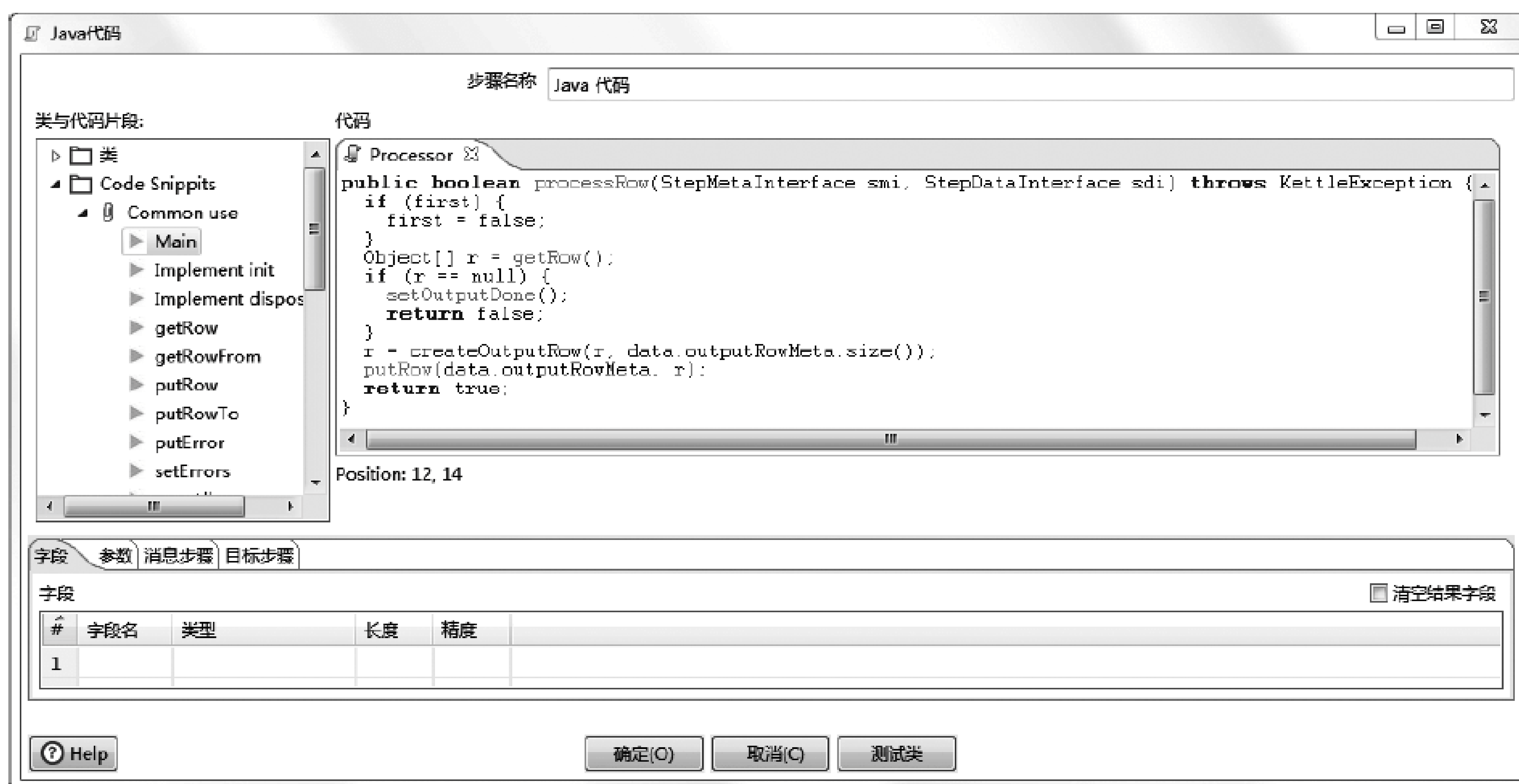


图 4-43 添加主方法的代码

在图 4-43 中的代码框编写抽取 HTML 网页数据的 Java 脚本代码，具体代码如文件 4-1 所示。

文件 4-1　Processor. class

```
1  import java.util.*;
2  import java.util.regex.Matcher;
3  import java.util.regex.Pattern;
4  import java.sql.DriverManager;
5  import java.sql.SQLException;
6  import java.util.regex.Matcher;
7  import java.util.regex.Pattern;
8  import com.mysql.jdbc.Connection;
9  import com.mysql.jdbc.PreparedStatement;
10 //声明输入的结果字段
11 private String result;
12 //声明输出的结果字段
13 private String contents;
14 //声明数据库连接
15 private Connection connection =null;
16 public boolean processRow(StepMetaInterface smi, StepDataInterface sdi)
17                                 throws KettleException,SQLException{
18     if (first) {
19         //获取传入的结果参数
20         result=getParameter("result");
21         first =false;
22     }
23     //获取一行记录,即抽取到的每行超链接数据
24     Object[] r =getRow();
25     if (r ==null) {
26         setOutputDone();
27         return false;
28     }
29     Object[] outputRow =createOutputRow(r, data.outputRowMeta.size());
30     String foobar =get(Fields.In, result).getString(r);
31     //编写抽取超链接<a></a>标签中数据的正则表达式规则
32     String pattern ="<a[^>]*href(\\\"([^\\\"]*)\\\"|\\'([^\\']*)\\'
33                                     |([^\\\\s>]*))[^>]*>(.*?)</a>";
34     //将匹配规则进行编译,生成 Pattern 对象,Pattern 对象表示编译后的正则表达式
35     Pattern patterns =Pattern.compile(pattern);
36     //将匹配规则与输入界面的 foobar 字段进行匹配,生成 Matcher 对象
37     Matcher m =patterns.matcher(foobar);
38     //查找与该匹配规则匹配的数据,并输出到数据库的 html 数据表中
39     while(m.find()){
40         //获取输出的字段,并进行赋值
41         get(Fields.Out, "contents").setValue(outputRow, m.group()
42                                     .replaceAll("<[^>]*>",""));
43         //配置数据库连接,用于保存抽取到的数据
44         String url ="jdbc:mysql://localhost:3306/extract";
45         String userName ="root";
46         String userPwd ="123456";
47             try{
48                 // 加载驱动程序
```

```
49                  Class.forName("com.mysql.jdbc.Driver");
50                  // 获取连接对象
51                  connection=(Connection) DriverManager.getConnection(url,
52                                                          userName, userPwd);
53                  } catch (Exception e) {
54                      e.printStackTrace();
55                  }
56          //要执行的 SQL 语句
57          String sql="insert into html (contents) values (?);";
58          PreparedStatement stat = (PreparedStatement) connection
59                                                  .prepareStatement(sql);
60          contents=m.group().replaceAll("<[^>]*>","");
61          stat.setString(1, contents);
62          stat.executeUpdate();
63          putRow(data.outputRowMeta, outputRow);
64      }
65      return true;
66 }
```

上述代码中，第 10～15 行代码声明输入的结果字段 result、输出的结果阶段 contents 和数据库连接 connection；第 16～28 行代码获取输入的结果字段和抽取到的超链接数据；第 30 行代码获取输入 UI(用户界面)的字段；第 32～37 行代码编写正则表达式，用于将输入 UI 的字段 foobar 与规则进行匹配；第 38～64 行代码查找与规则匹配的数据，并将匹配到的数据保存至数据库的 html 数据表中。

单击图 4-43 中的“字段”选项卡，用于添加新生成的字段；单击“参数”选项卡，用于传入参数。“字段”选项卡界面和“参数”选项卡界面如图 4-44 所示。

图 4-44 “字段”选项卡界面和“参数”选项卡界面

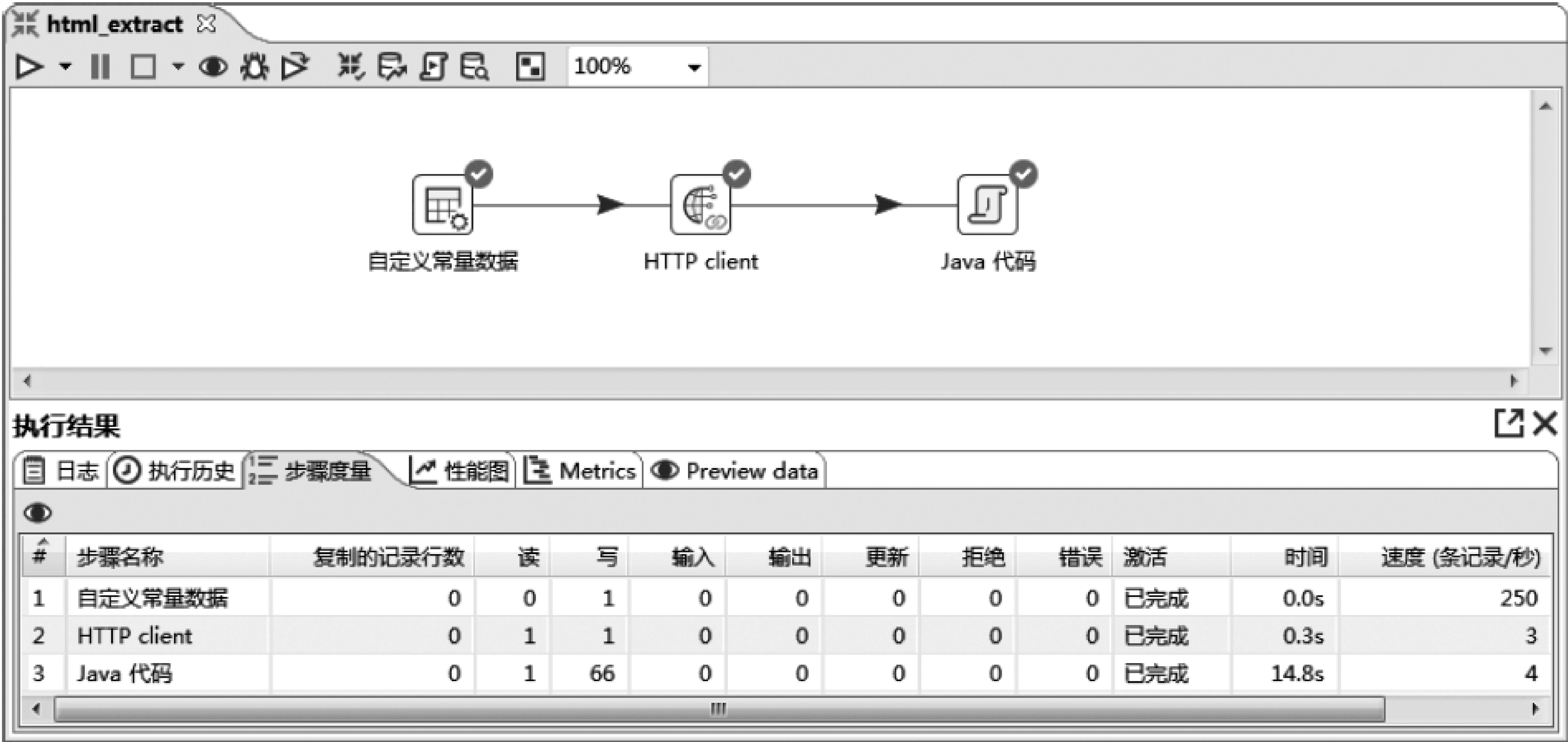
在图 4-44 中的“字段”选项卡界面添加新生成的字段 contents，并指定数据类型 String；在“参数”选项卡界面传入参数 result，并指定值为 result，具体如图 4-45 所示。

图 4-45 配置生成字段 contents 和传入参数 result

单击图 4-45 中的“确定”按钮完成“Java 代码”控件的配置。

5. 运行转换 html_extract

单击转换工作区顶部的▷按钮，运行创建的转换 html_extract，实现抽取 HTML 网页的数据，并将数据保存到数据表 html 中，具体如图 4-46 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活	时间	速度（条记录/秒）
1	自定义常量数据	0	0	1	0	0	0	0	0	已完成	0.0s	250
2	HTTP client	0	1	1	0	0	0	0	0	已完成	0.3s	3
3	Java 代码	0	1	66	0	0	0	0	0	已完成	14.8s	4

图 4-46　运行转换 html_extract

从图 4-46 中执行结果的“步骤度量”可以看出，“自定义常量数据”控件写入 1 条记录；HTTP client 控件读取“自定义常量数据”控件中的 1 条记录并写入该控件中；“Java 代码”控件读取 HTTP client 控件中的 1 行记录，写入该控件 66 条数据，并写入数据表 html 中。

6. 查看数据表 html 中的数据

通过 SQLyog 工具，查看数据表 html 是否已成功插入 66 条数据，查看结果如图 4-47 所示。

从图 4-47 中可以看出，数据表 html 中已插入数据(这里只展示数据表中的部分数据)，说明我们成功实现了将抽取到的 HTML 网页数据保存到数据表 html 中。

contents
登录/注册
下载豆瓣客户端
豆瓣 6.0 全新发布
×
iPhone
Android
豆瓣
读书
电影
音乐
同城
小组
阅读
FM
时间
豆品

图 4-47　数据表 html

4.2.2　XML 文件的数据抽取

XML 是一种可扩展标记语言，也是一种元标记语言。所谓“元标记”，就是开发者可根据自己的需要自定义标记。XML 是一种很像 HTML 的标记语言，但是它们也有很大的区别，如 XML 被设计出来，主要用于传输和存储数据，其焦点是数据的内容，而 HTML 被设计出来，主要用于显示数据，其焦点是数据的外观；XML 中的标签是没有被预定义的，都是由 XML 文档的创作者发明的，HTML 中的标签是预定义的，其文档中使用的标签必须是在 HTML 标准中定义过的，用户自己定义的标签是不可使用的。

XML 应用于 Web 开发的多个方面，但常用于简化数据的存储和共享。下面介绍一下

XML 的主要用途。

- XML 可以将数据从 HTML 中分离。

如果要在 HTML 文档中显示动态数据，那么每当数据改变时，都需要花费大量时间编辑 HTML，这样效率很低。使用 XML 可以将数据存储在独立的 XML 文件中，通过编程读取一个外部的 XML 文件，并更新网页的数据内容。

- XML 可以简化数据传输。

通过 XML 可在不兼容的系统之间传输数据。现实生活中，计算机系统和数据库系统存储的数据有 N^N 种形式，对于开发者来说，最耗时间的是在遍布网络的系统之间传输数据，将数据转换为 XML 格式进行存储，大大降低了交换数据的复杂性，也便于数据被不同的程序读取。

- XML 可以简化数据共享。

XML 数据是以纯文本格式存储的，因此 XML 提供了一种独立于软件和硬件的数据存储方法，便于不同应用程序共享数据。

- XML 简化平台变更。

升级操作系统、服务器、应用程序以及更新浏览器是非常浪费时间的。例如，转换大量的数据经常会导致不兼容的数据丢失，而 XML 数据是以文本格式存储的，这使得 XML 在不损失数据的情况下更容易扩展或升级到新的操作系统、应用程序以及浏览器。

- XML 可以使数据充分利用。

不同的应用程序可以从 HTML 页面中访问数据，也可以从 XML 数据源中访问数据，使得数据供各种阅读设备(如掌上计算机、语音设备、新闻阅读器等)使用。

- XML 可用于存储数据。

由于纯文本文件可用来存储数据，因此可以将大量的数据存储到 XML 文件中或数据库中。应用程序可对数据进行读写和存储，然后通过编写程序显示数据。

- XML 可用于创建新的互联网语言。

互联网语言中有很多新的语言是通过 XML 创建的，如 XHTML、WSDL(用于描述可用的 Web 服务)和 WAP(手持设备的标记语言)。

现有一个 XML 文件，名为 xml_extract.xml，具体内容如图 4-48 所示。

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <AllRows testDescription="1 - simple functionality test">
3      <Rows rowID="1">first row chunk of data
4          <Row><v1>1.1.1</v1><v2>1.1.2</v2></Row>
5          <Row><v1>1.2.1</v1><v2>1.2.2</v2></Row>
6      </Rows>
7      <Rows rowID="2">second row chunk of data
8          <Row><v1>2.1.1</v1><v2>2.1.2</v2></Row>
9          <Row><v1>2.2.1</v1><v2>2.2.2</v2></Row>
10     </Rows>
11     <Rows rowID="3">third row chunk of data
12         <Row><v1>3.1.1</v1><v2>3.1.2</v2></Row>
13         <Row><v1>3.2.1</v1><v2>3.2.2</v2></Row>
14     </Rows>
15 </AllRows>
```

图 4-48　文件 xml_extract.xml 的内容

下面分步骤讲解如何抽取 XML 文件 xml_extract. xml 中的数据并保存至数据库 extract 的数据表 xml 中，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 xml_extract，并添加 Get data from XML 控件、“表输出”控件以及 Hop 跳连接线，用于实现抽取 XML 文件中标签 testDescription、rowID、v1 以及 v2 中的数据，并保存至数据表 xml 中，具体效果如图 4-49 所示。

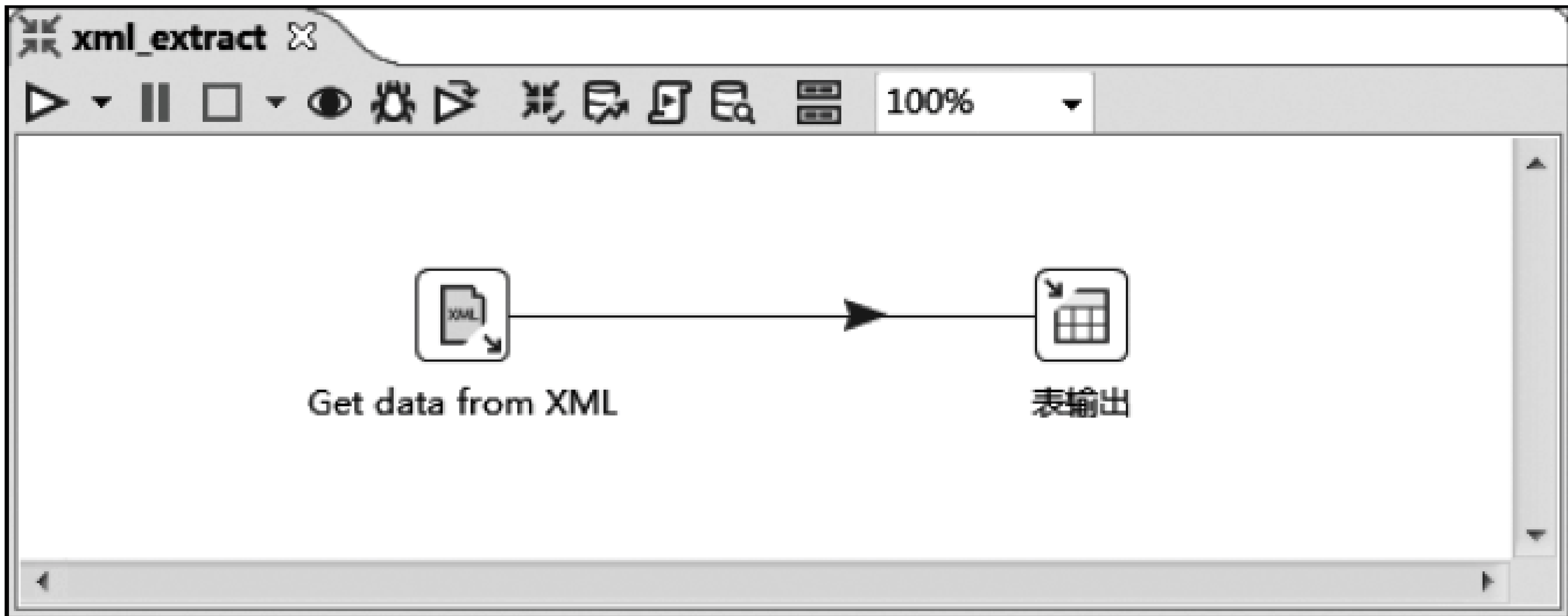


图 4-49　创建转换 xml_extract

2. 配置 Get data from XML 控件

双击图 4-49 中的 Get data from XML 控件，进入“XML 文件输入”界面，具体如图 4-50 所示。



图 4-50　“XML 文件输入”界面

单击图 4-50 中的“浏览”按钮，选择要抽取的 XML 文件 xml_extract. xml，具体如图 4-51 所示。

单击图 4-51 中的“增加”按钮，将所选择的文件路径添加到“选中的文件和目录”处，具

体如图 4-52 所示。

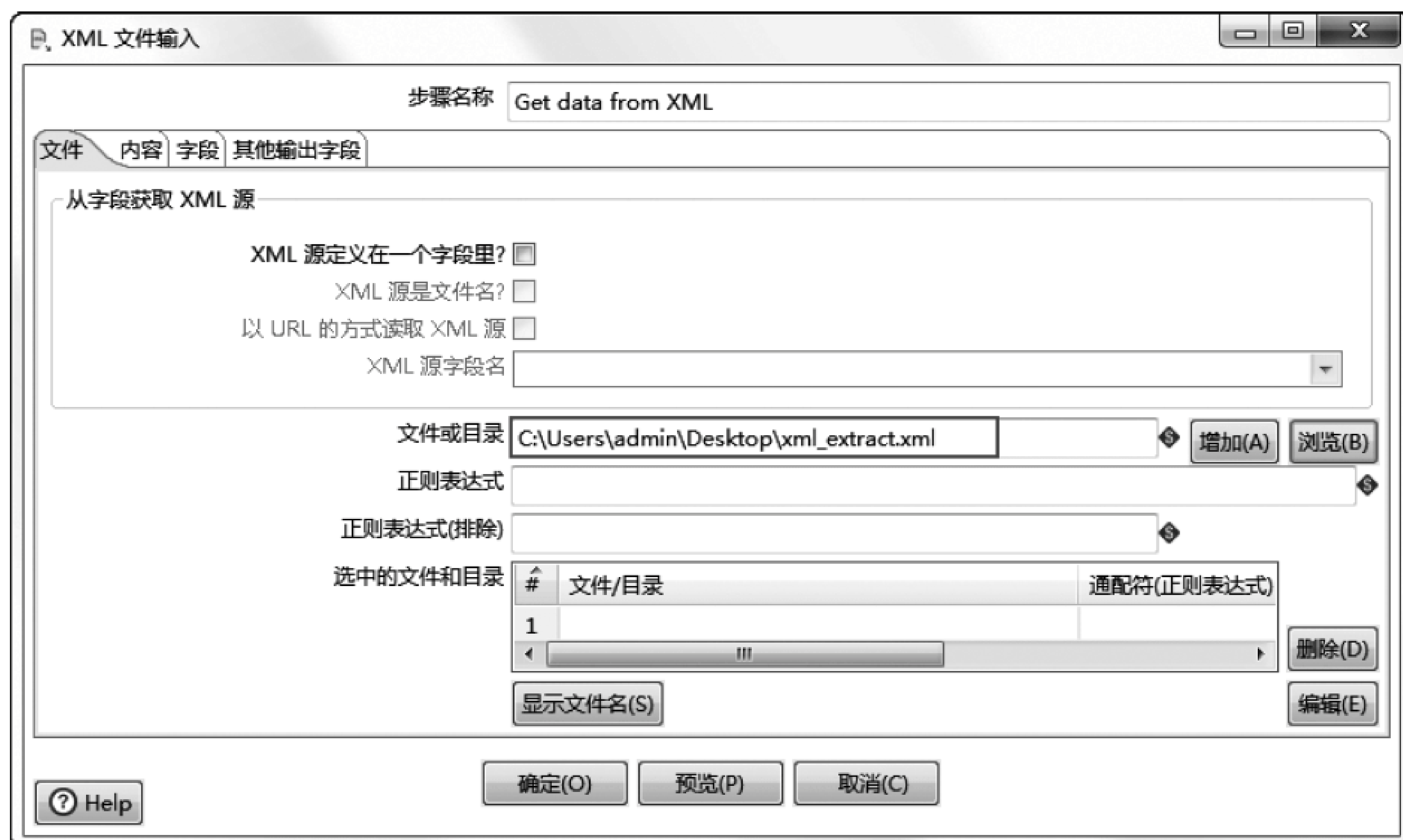


图 4-51　选择要抽取的 XML 文件 xml_extract. xml



图 4-52　添加 XML 文件 xml_extract. xml 至“选中的文件和目录”处

单击图 4-52 中“内容”选项卡，进入“内容”选项卡界面，具体如图 4-53 所示。

单击图 4-53 中的“获取 XML 文档的所有路径”添加循环读取路径，即/AllRows/Rows/Row，具体效果如图 4-54 所示。

单击图 4-54 中的“字段”选项卡，进入“字段”选项卡界面，具体如图 4-55 所示。

在图 4-55 中添加要抽取的字段，具体如图 4-56 所示。



图 4-53　“内容”选项卡界面



图 4-54　添加循环读取路径

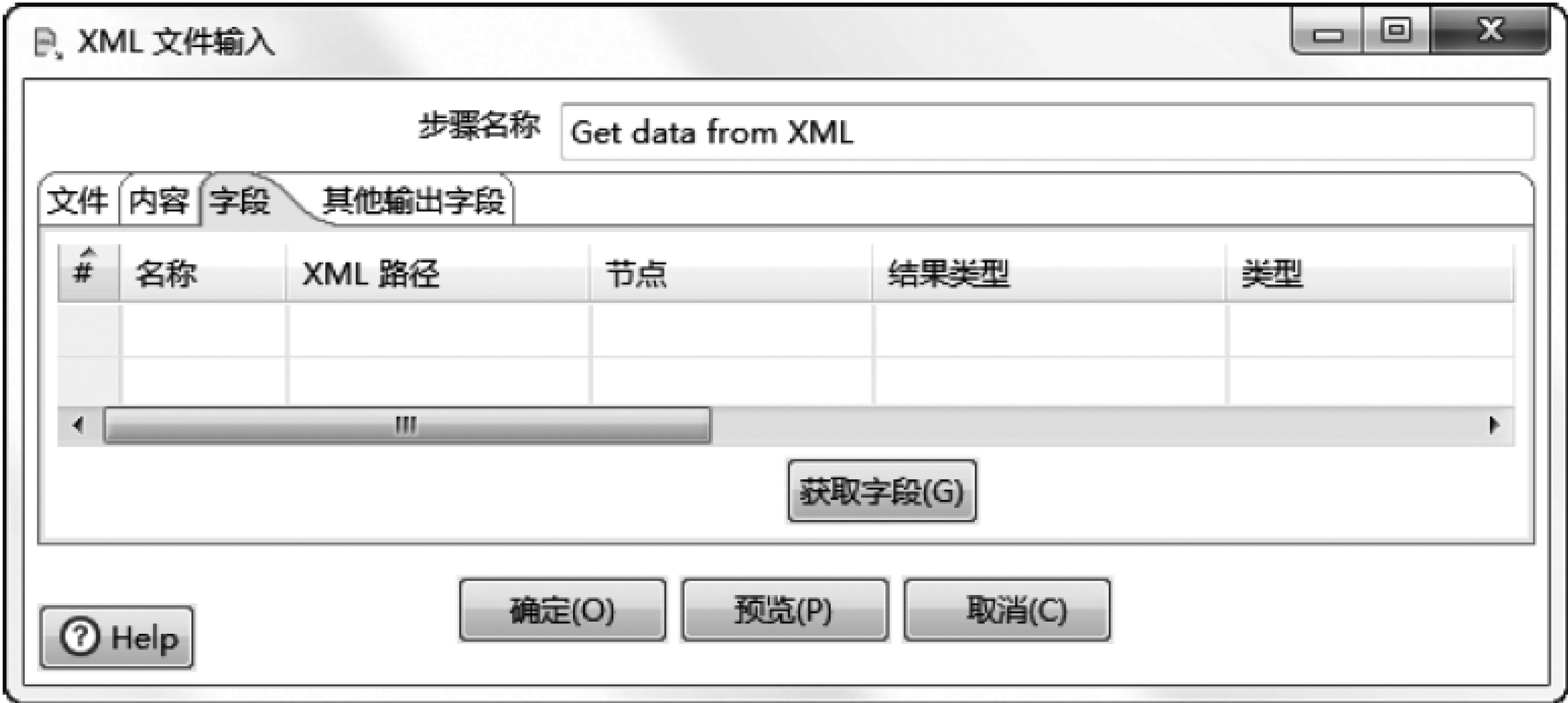


图 4-55　“字段”选项卡



图 4-56　添加要抽取的字段

单击图 4-56 中的“确定”按钮，完成 Get data from XML 控件的配置。

3. 配置“表输出”控件

双击图 4-49 中的“表输出”控件，进入“表输出”界面，具体如图 4-57 所示。



图 4-57　“表输出”界面

单击图 4-57 中的“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置具体如图 4-58 所示。

单击图 4-57 中目标表右侧的“浏览”按钮，选择输出的目标表，即 xml 数据表(该表需提前创建，且表结构需根据 XML 文件 xml_extract.xml 中的数据字段和数据类型进行创建，这里不演示)；勾选“指定数据库字段”复选框，用于将数据表 xml 的字段与 XML 文件 xml_extract.xml 中的字段进行匹配，具体如图 4-59 所示。

单击图 4-59 中的“数据库字段”选项卡，进入“数据库字段”选项卡界面，具体如图 4-60 所示。

单击图 4-60 中的“输入字段映射”按钮，弹出“映射匹配”对话框，如图 4-61 所示。

在图 4-61 中依次选中“源字段”中的字段和“目标字段”中对应的字段，单击 Add 按钮，

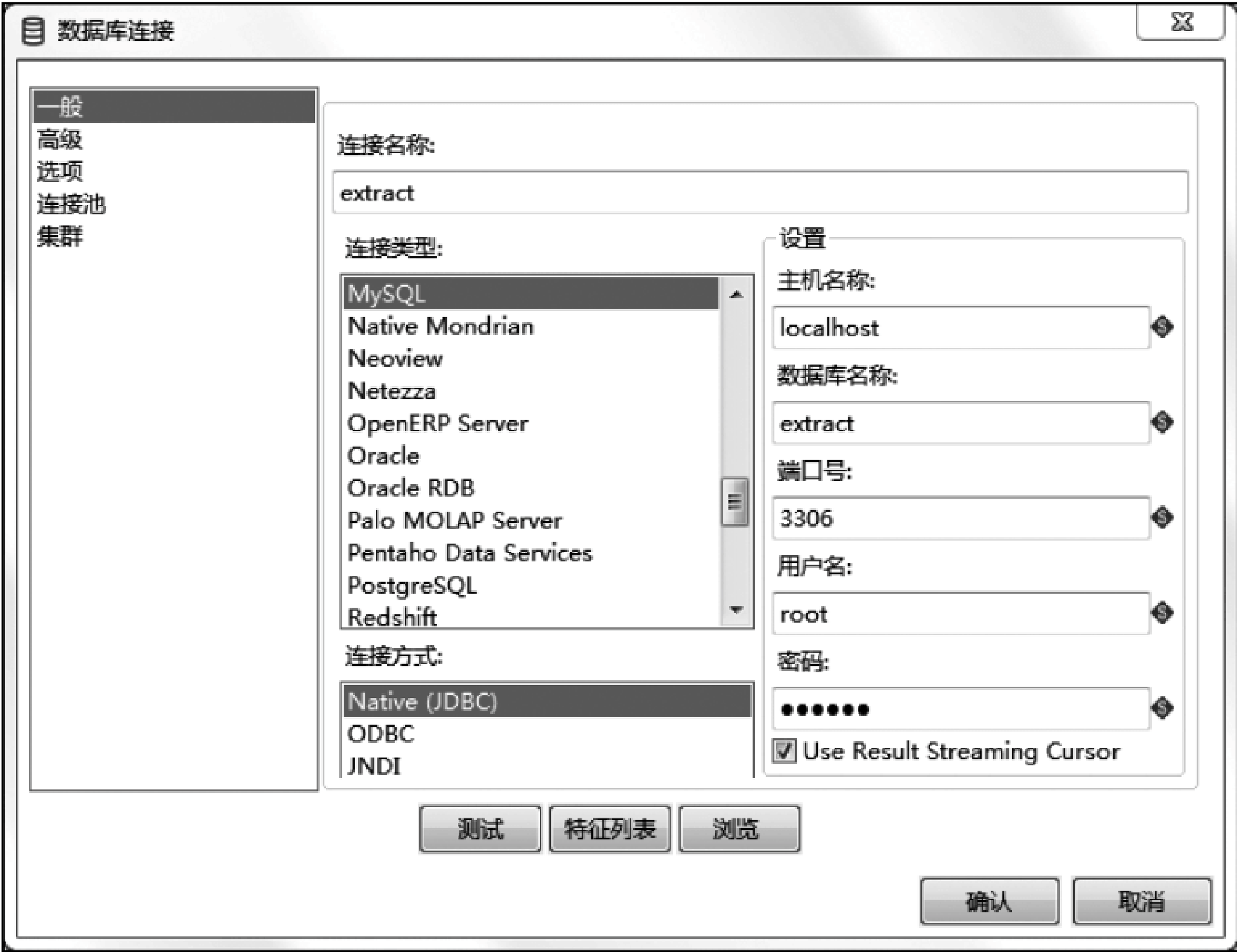


图 4-58　MySQL 数据库连接的配置



图 4-59　指定输出的目标表

将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 4-62 所示。

单击图 4-62 中的“映射匹配”对话框中的“确定”按钮，“表输出”界面最终显示的效果如图 4-63 所示，单击“确定”按钮，完成“表输出”控件的配置。

4．运行转换 xml_extract

单击转换工作区顶部的▷按钮，运行创建的转换 xml_extract，实现将 XML 文件中的数据抽取到数据表 xml 中，具体如图 4-64 所示。



图 4-60 “数据库字段”选项卡

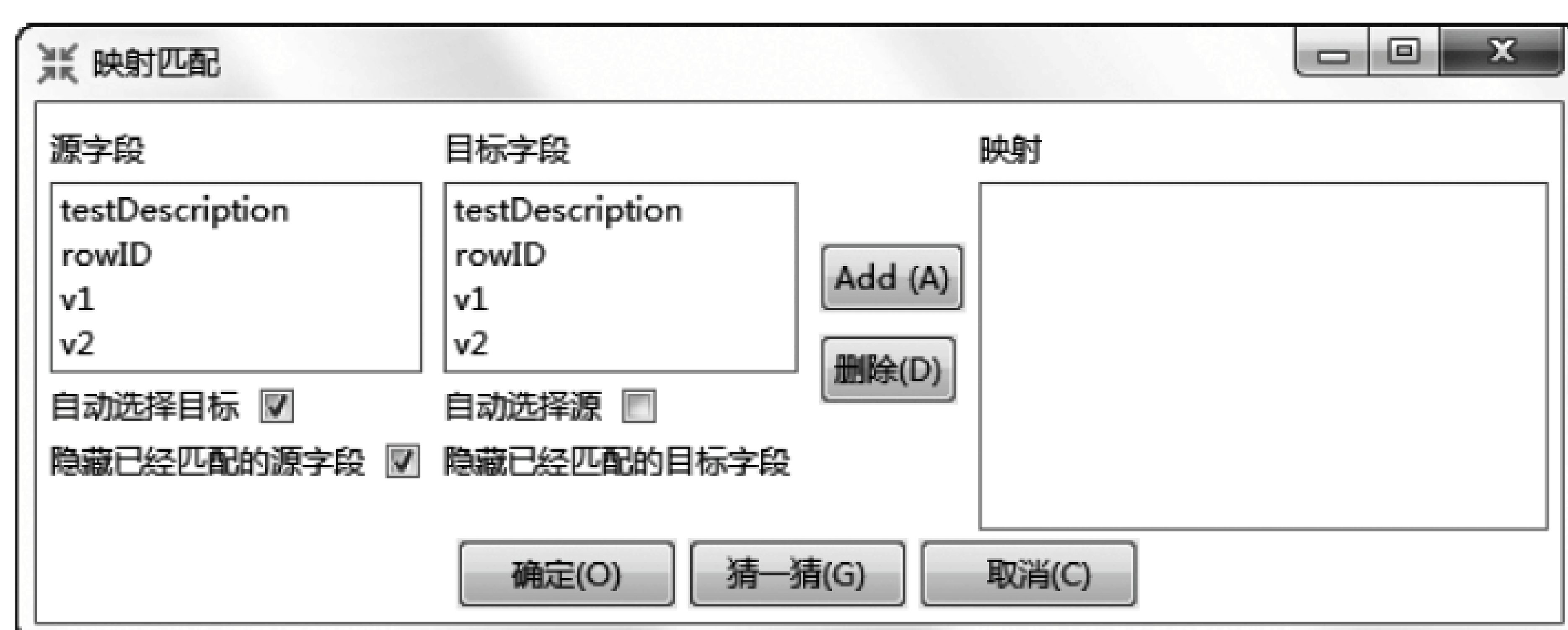


图 4-61 “映射匹配”对话框

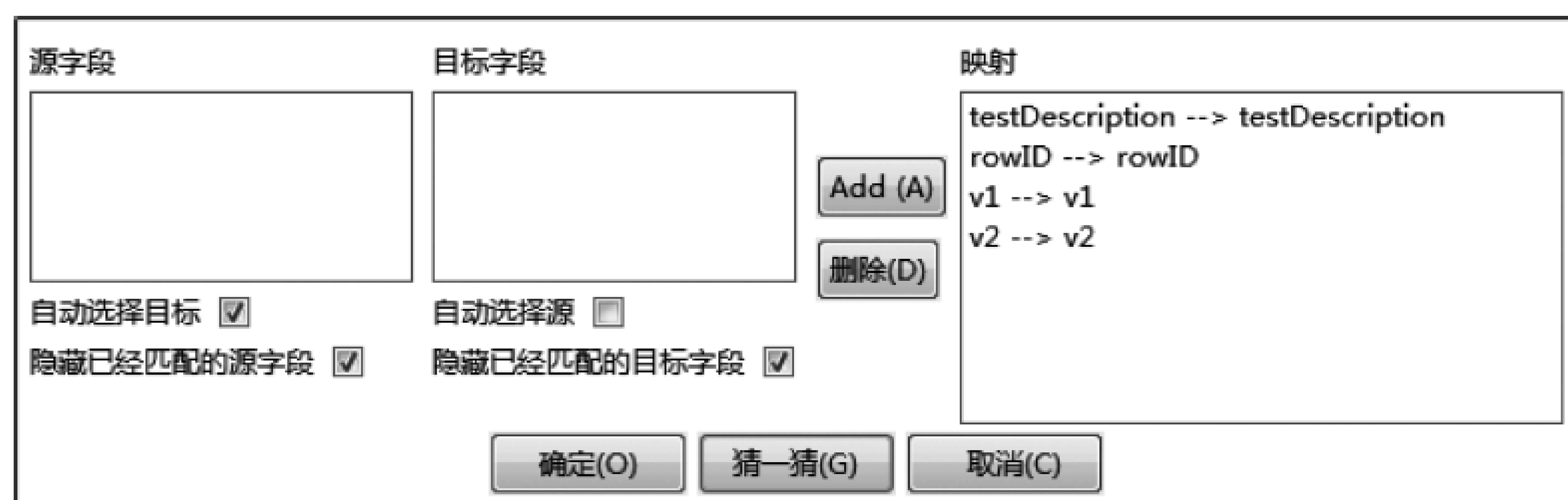


图 4-62 设置映射匹配

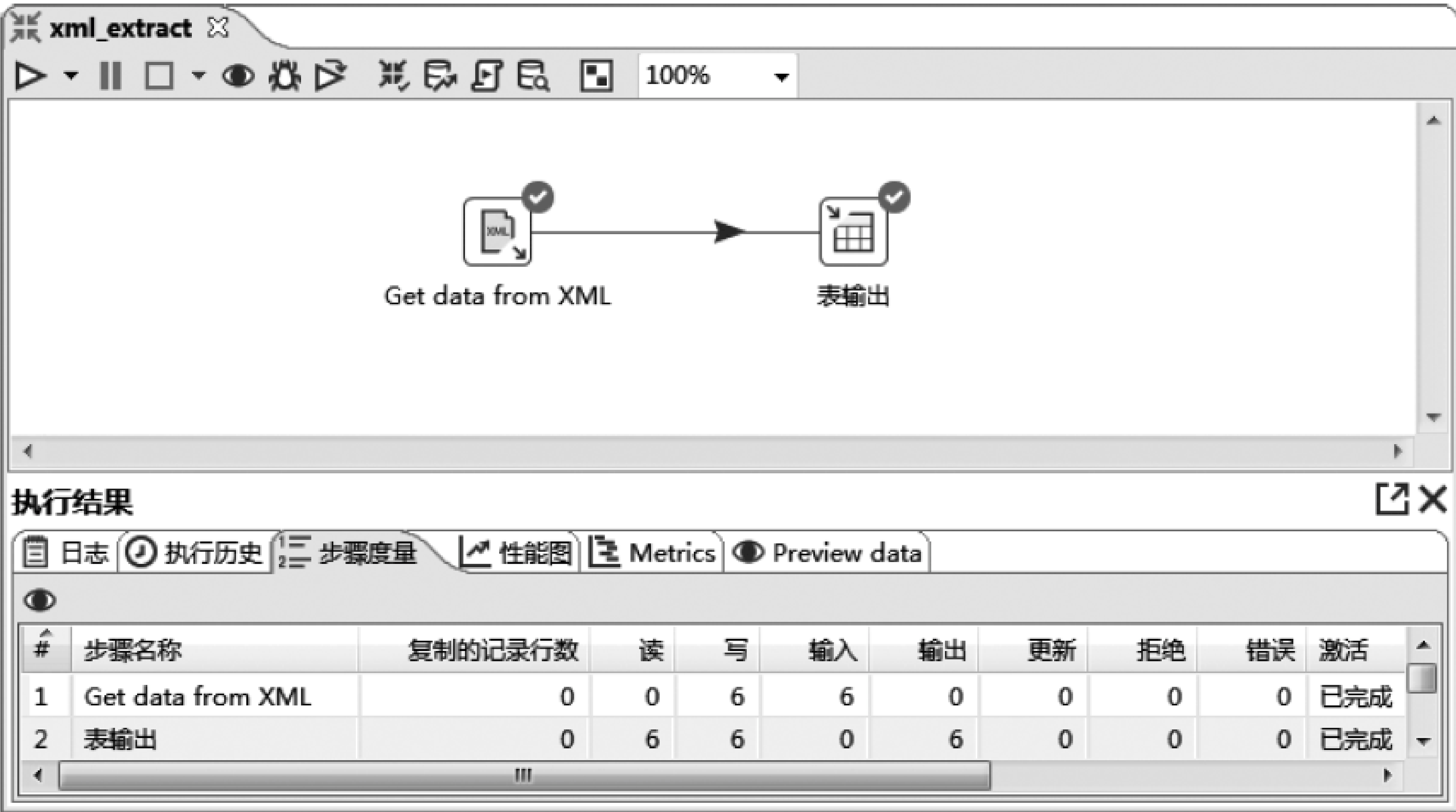
从图 4-64 中执行结果的“步骤度量”可以看出，Get data from XML 控件写入 6 条输入并输入该控件，而“表输出”控件读取 Get data from XML 控件中的 6 条数据并写入该控件，最终进行输出。也就是说，“表输出”控件将从 Get data from XML 流中读取的 6 条数据写入数据表 xml 中。

5. 查看数据表 xml 中的数据

通过 SQLyog 工具，查看数据表 xml 是否已成功插入 6 条数据，查看结果如图 4-65 所示。

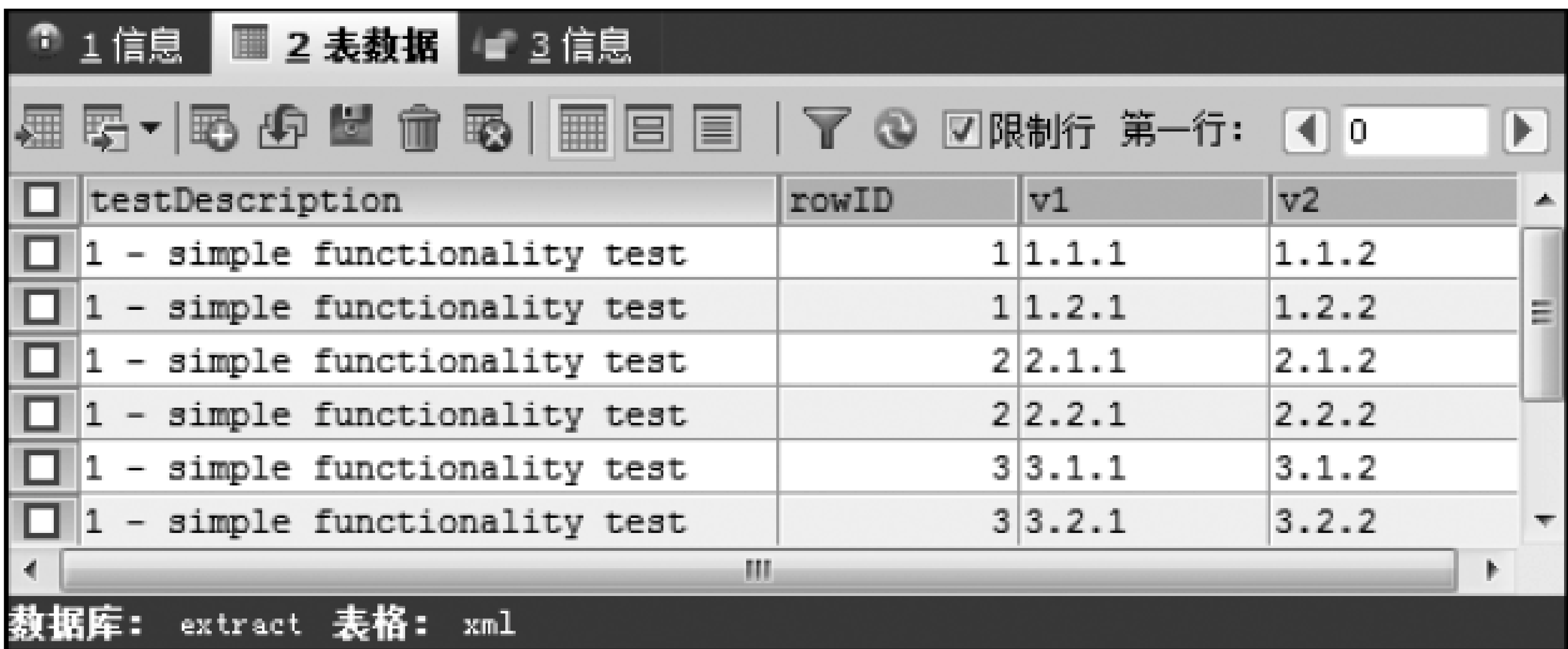


图 4-63　“表输出”界面最终显示的效果



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	Get data from XML	0	0	6	6	0	0	0	0	已完成
2	表输出	0	6	6	0	6	0	0	0	已完成

图 4-64　运行转换 xml_extract



testDescription	rowID	v1	v2
1 - simple functionality test	1	1.1.1	1.1.2
1 - simple functionality test	1	1.2.1	1.2.2
1 - simple functionality test	2	2.1.1	2.1.2
1 - simple functionality test	2	2.2.1	2.2.2
1 - simple functionality test	3	3.1.1	3.1.2
1 - simple functionality test	3	3.2.1	3.2.2

数据库： extract　表格： xml

图 4-65　数据表 xml

从图 4-65 中可以看出，数据表 xml 中已插入数据，说明我们成功实现了将 XML 文件 xml_extract. xml 中标签 testDescription、rowID、v1 以及 v2 中的数据抽取到数据表 xml 中。

4.2.3 JSON 文件的数据抽取

JSON(JavaScript Object Notation，JS 对象标记)是一种轻量级的数据交换格式。它是基于 ECMAScript (欧洲计算机协会制定的 js 规范)的一个子集，从 JavaScript 脚本语言中演变而来，采用完全独立于编程语言的文本格式存储和表示数据。由于 JSON 有简洁、清晰的层次结构，因此使得 JSON 成为理想的数据交换语言。JSON 易于程序开发者阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输的效率。

需要注意的是，JSON 是一种文本数据交换格式，而并非编程语言，其语法只支持字符串、数字(整数或浮点数)、布尔值、null 以及对象和数组等类型。常用的类型是对象和数组。对象是使用花括号(即{})括起来的内容，数据结构为{key1：value1，key2：value2，…}，在面向对象的语言中，key 为对象的属性，value 为对象的值，键名可用整数和字符串表示，值的类型可以是任意类型；数组是使用方括号(即[])括起来的内容，数据结构为["java"，"javascript"，...]的索引结构，值的类型可以是任意类型。

现有一个 JSON 文件，名为 json_extract. json，部分内容如图 4-66 所示。

```
{
    "status": "ok",
    "response": {
        "submissions": [
            {
                "id": "59434767",
                "timestamp": "2011-11-21 09:21:53",
                "user_agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/8.0",
                "remote_addr": "192.168.1.1",
                "payment_status": "",
                "data": [
                    {
                        "field": "13776121",
                        "value": "Baylor Dallas"
                    },
                    {
                        "field": "13776401",
                        "value": "CHF"
                    },
                    {
                        "field": "13777966",
                        "value": "John Doe"
                    }
                ]
            },
            {
        ],
        "total": 2,
        "pages": 1
    }
}
```

图 4-66 文件 json_extract. json 的部分内容

下面分步骤讲解如何抽取 JSON 文件 json_extract. json 中的数据并保存至数据库 extract 中的数据表 json 中，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 json_extract，并添加 JSON input 控件、“表输出”控件以及 Hop 跳连接线，用于实现抽取 JSON 文件中 key 值为 id、field 和 value 的数据，并保存至数据表 json 中，具体效果如图 4-67 所示。

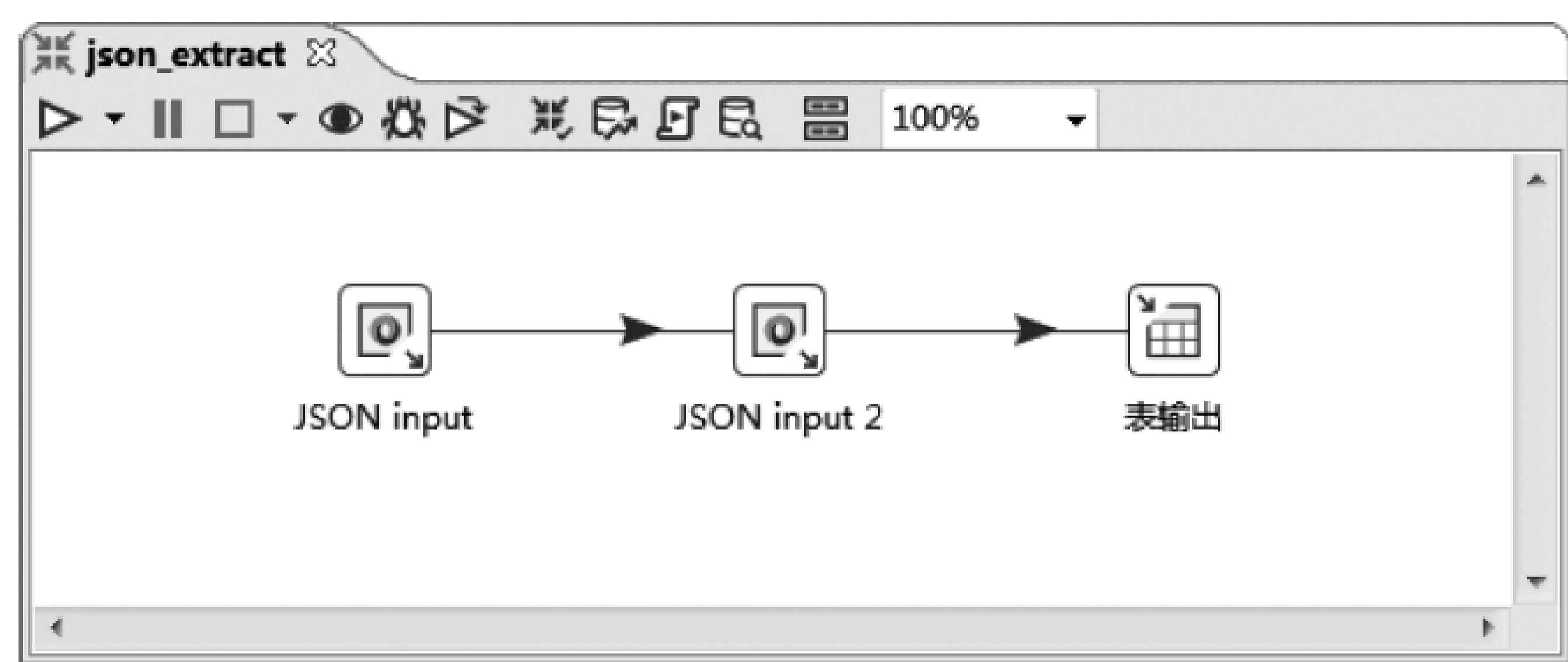


图 4-67　创建转换 json_extract

2. 配置 JSON input 控件

双击图 4-67 中的 JSON input 控件,进入“JSON 输入”界面,如图 4-68 所示。



图 4-68　“JSON 输入”界面

在图 4-68 中单击“浏览”按钮,选择要抽取的 JSON 文件 json_extract. json,如图 4-69 所示。

在图 4-69 中单击“增加”按钮,将所选择的文件添加到“选中的文件”处,具体效果如图 4-70 所示。

在图 4-70 中单击“字段”选项卡,进入“字段”选项卡界面,如图 4-71 所示。

在图 4-71 中添加要抽取的数据字段(这里采用分层抽取数据字段,先抽取 id 和 data 字段,再从 data 字段中抽取 field 和 value 字段),具体配置如图 4-72 所示。

在图 4-72 中单击“确定”按钮,完成 JSON input 控件的配置。

双击图 4-67 中的 JSON input 2 控件,进入“JSON 输入”界面,如图 4-73 所示。

在图 4-73 中勾选“源定义在一个字段里?”复选框;在“从字段获取源”后的下拉列表中



图 4-69 选择要抽取的 json_extract.json 文件



图 4-70 添加文件 json_extract.json

选择字段名，即 data，具体如图 4-74 所示。

在图 4-74 中单击“字段”选项卡，进入“字段”选项卡界面，如图 4-75 所示。

在图 4-75 中添加从字段 data 中抽取的 field 和 value 字段，具体配置如图 4-76 所示。

在图 4-76 中单击“确定”按钮，完成 JSON input 2 控件的配置。



图 4-71　“字段”选项卡界面

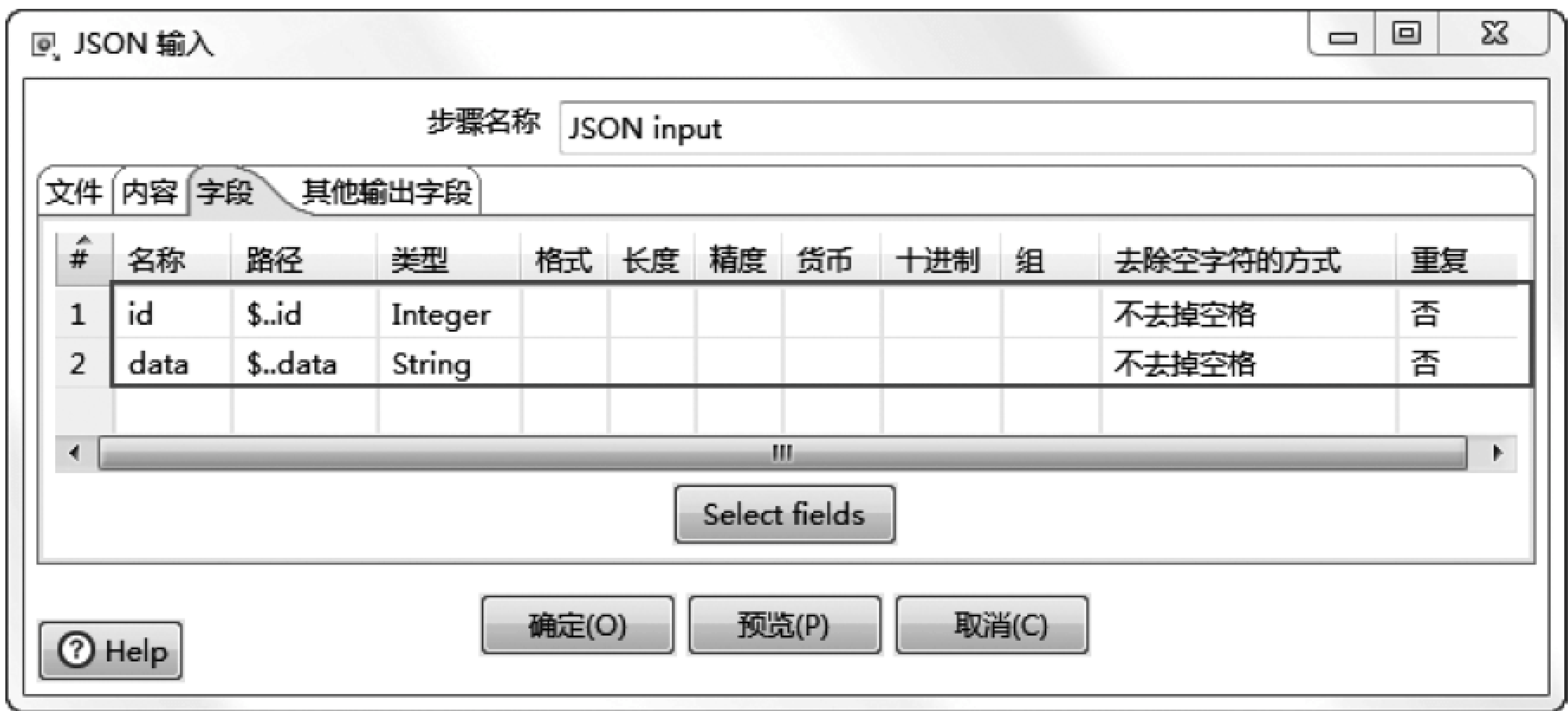


图 4-72　抽取 id 和 data 字段的配置



图 4-73　“JSON 输入”界面



图 4-74　从字段获取源的配置

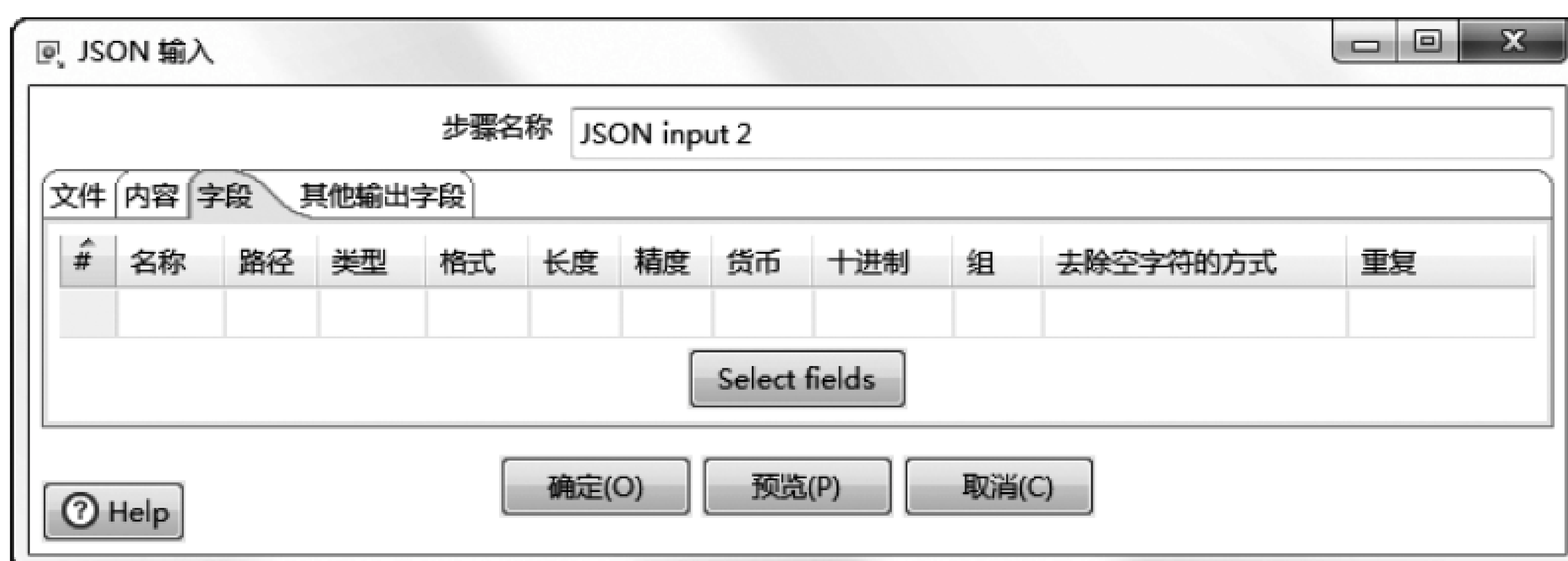


图 4-75　“字段”选项卡界面

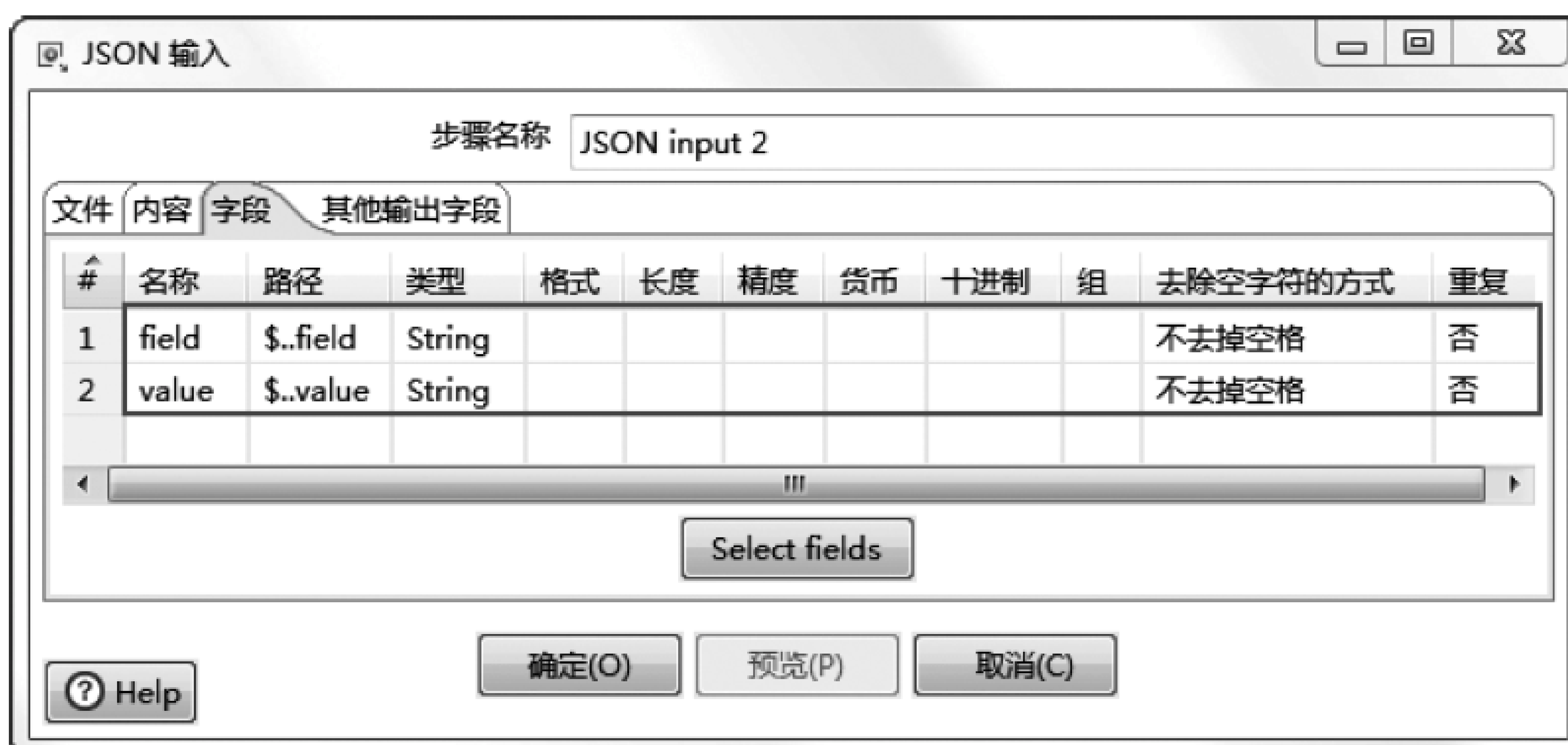


图 4-76　配置抽取的 field 和 value 字段

3. 配置“表输出”控件

双击图 4-67 中的“表输出”控件，进入“表输出”界面，具体如图 4-77 所示。



图 4-77　“表输出”界面

在图 4-77 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 4-78 所示。

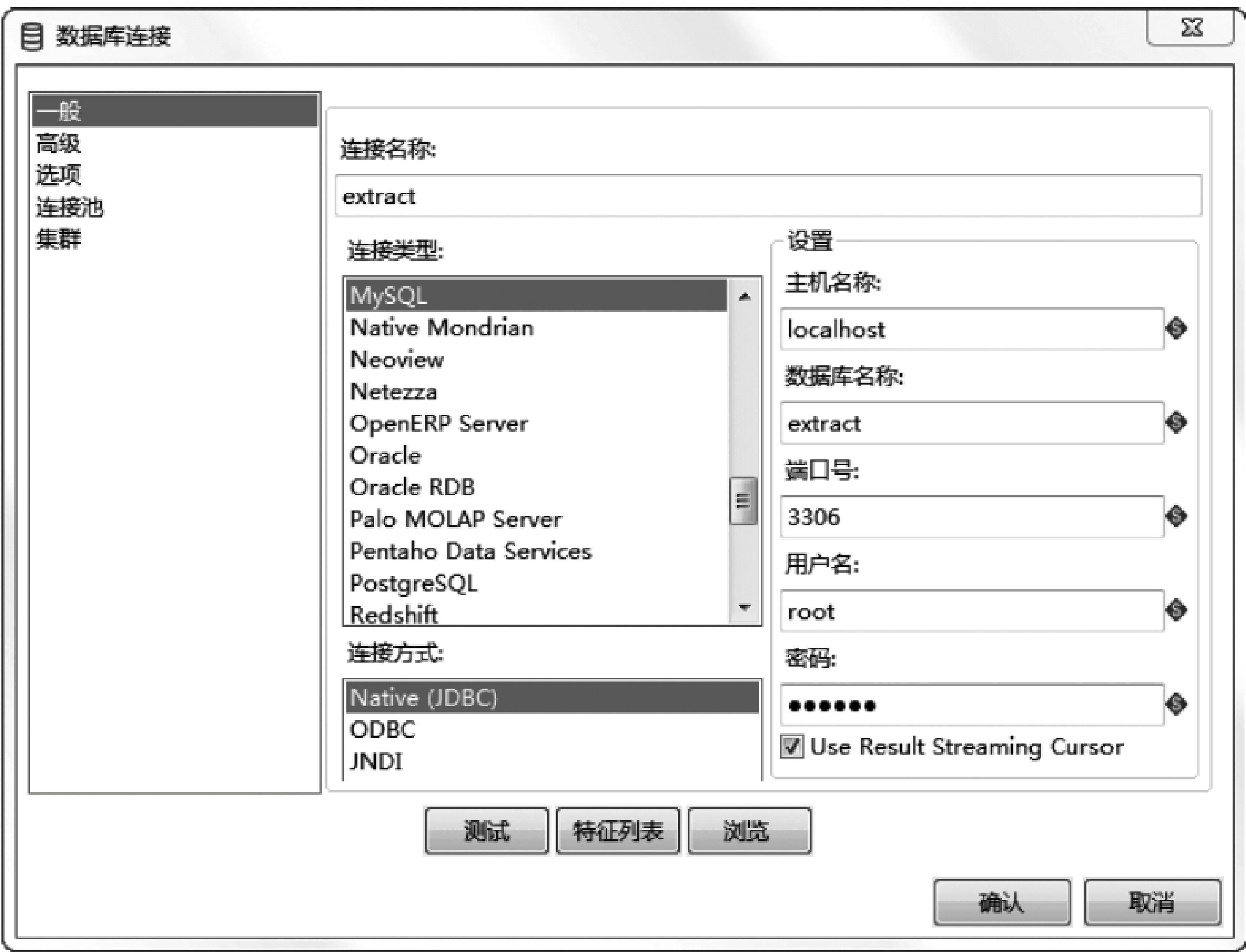


图 4-78　MySQL 数据库连接的配置

单击图 4-77 中目标表右侧的“浏览”按钮，选择输出的目标表，即数据表 json（该表需提前创建，且表结构需根据 json_extract.json 中的数据字段和数据类型进行创建，这里不演示）；勾选“指定数据库字段”复选框，用于将数据表 json 的字段与 JSON 文件 json_extract.json 中的字段进行匹配，具体如图 4-79 所示。



图 4-79　指定输出的目标表

在图 4-79 中单击“数据库字段”选项卡，进入“数据库字段”选项卡界面，如图 4-80 所示。



图 4-80　“数据库字段”选项卡界面

在图 4-80 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，如图 4-81 所示。



图 4-81　“映射匹配”对话框

在图 4-81 中依次选中“源字段”中的字段和“目标字段”中对应的字段，单击 Add 按钮，将一对映射字段添加至“映射”框中。若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 4-82 所示。



图 4-82　设置映射匹配

在图 4-82 中单击“映射匹配”对话框中的“确定”按钮，“表输出”界面最终显示的效果如图 4-83 所示。

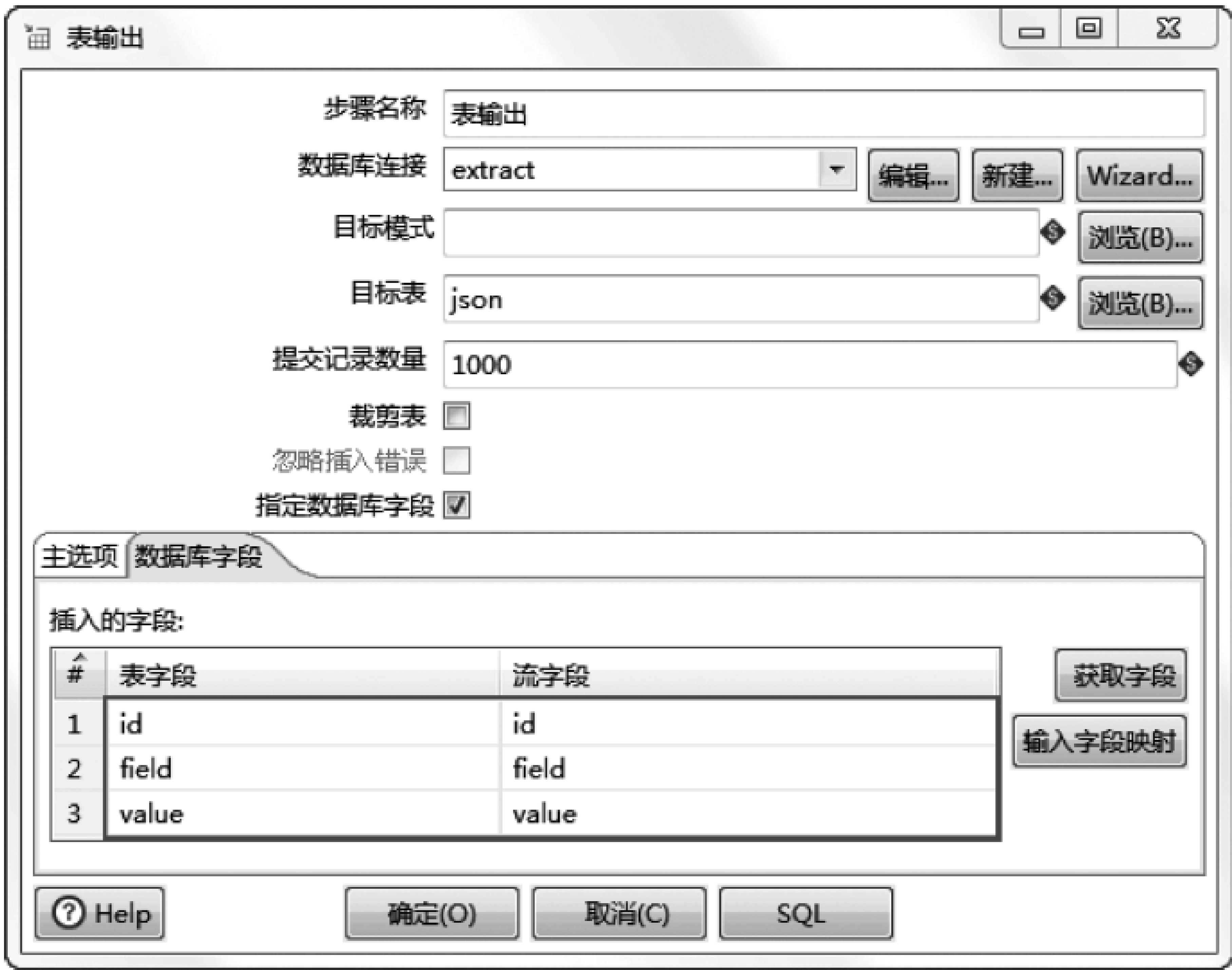


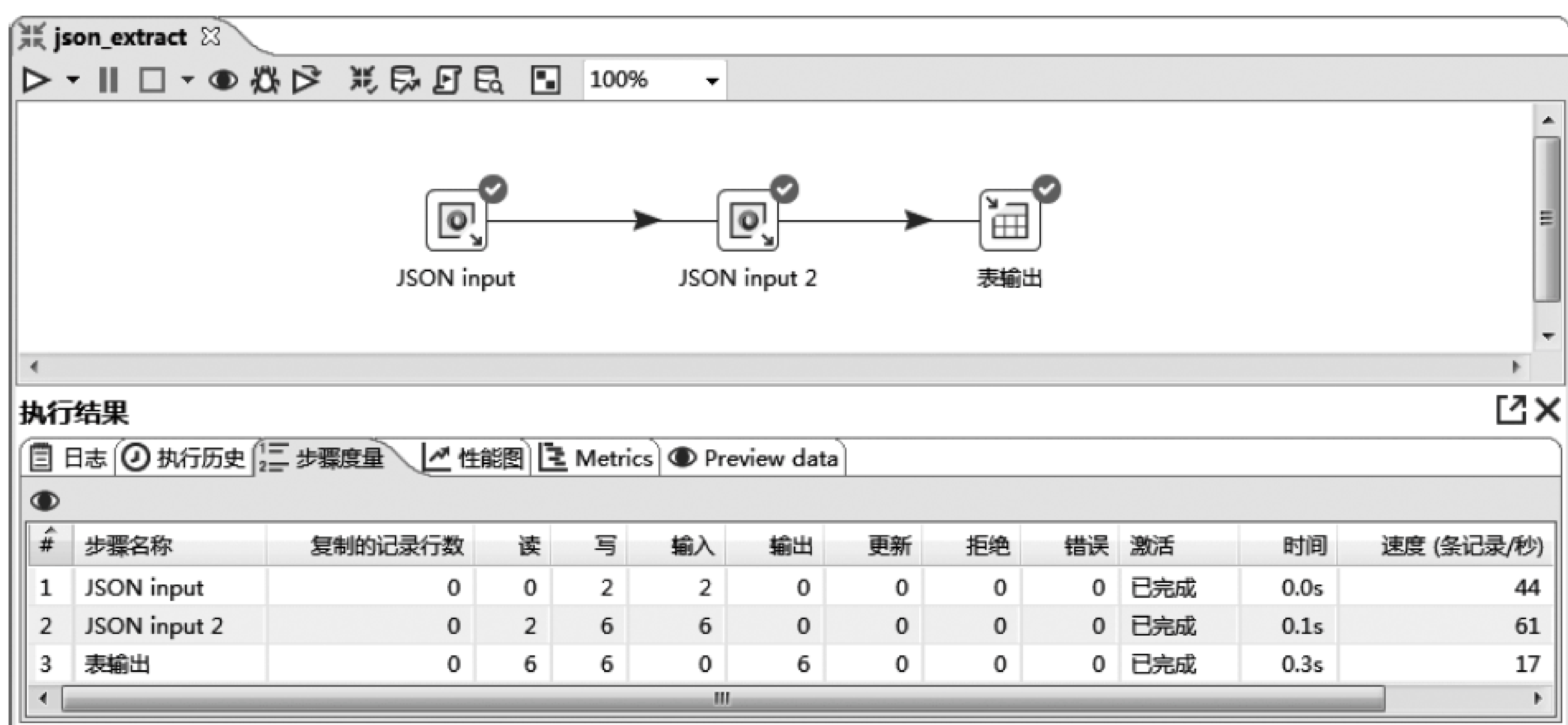
图 4-83　“表输出”界面最终显示的效果

在图 4-83 中单击“确定”按钮，完成“表输出”控件的配置。

4. 运行转换 json_extract

单击转换工作区顶部的▷按钮，运行创建的转换 json_extract，实现将 JSON 文件中的数据抽取到数据表 json 中，具体如图 4-84 所示。

从图 4-84 中执行结果的“步骤度量”可以看出，JSON input 控件输入 2 条数据并写入该控件中；JSON input 2 控件读取 JSON input 控件的 2 条数据，从这 2 条数据的 data 字段中共获取 6 条数据作为输入并写入该控件中；“表输出”控件读取 JSON input 控件的两条数据，从这两条数据的 data 字段中共获取 6 条数据作为输入并写入该控件中。也就是说，“表



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活	时间	速度 (条记录/秒)
1	JSON input	0	0	2	2	0	0	0	0	已完成	0.0s	44
2	JSON input 2	0	2	6	6	0	0	0	0	已完成	0.1s	61
3	表输出	0	6	6	0	6	0	0	0	已完成	0.3s	17

图 4-84　运行转换 json_extract

输出"控件从 JSON input 2 流中读取的 6 条数据均写入数据表 json 中。

5. 查看数据表 json 中的数据

通过 SQLyog 工具，查看数据表 json 是否已成功插入 6 条数据，查看结果如图 4-85 所示。

1 信息　2 表数据　3 信息

id	field	value
59434767	13776121	Baylor Dallas
59434767	13776401	CHF
59434767	13777966	John Doe
59474875	13776121	Healthsouth,
59474875	13776401	Pneumonia
59474875	13777966	Jane Doe

数据库: extract　表格: json

图 4-85　数据表 json

从图 4-85 中可以看出，数据表 json 中已插入数据，说明我们成功实现了将 JSON 文件 json_extract.json 中 key 值为 id、field 和 value 的数据抽取到数据表 json 中。

4.3　抽取数据库数据

数据库(Database)是按照数据结构组织、存储和管理数据的仓库。在信息化社会，充分有效地管理和利用各类信息资源，是进行科学研究和决策管理的前提条件。数据库共有两种类型，分别是关系型数据库和非关系型数据库。其中关系型数据库有 MySQL、Oracle 及 SQL Server 等数据库，非关系型数据库有 MongoDB、Redis 及 HBase 等数据库。本节将对关系型数据库和非关系型数据库的数据抽取分别进行详细讲解。

4.3.1　抽取关系型数据库的数据

在传统的大型企业中，业务系统大多采用 Microsoft SQL Server 数据库存储数据，而数

据仓库的缓存层则采用 MySQL 数据库缓存数据。在实际工作中,我们会通过数据仓库对业务系统产生的数据进行分析,因此需要将业务系统的数据从 Microsoft SQL Server 数据库抽取到数据仓库缓存层(即 MySQL 数据库)中,然后进行相关的数据分析。

现有一个数据表,名为 student(存在于 Microsoft SQL Server 数据库中,需读者提前创建),具体内容如图 4-86 所示。

id	name	age	sex
s001	Allen	18	female
s002	Linda	20	female
s003	Eric	18	male
s004	Lily	18	female
s005	Rose	19	female
s006	Abbott	22	male
s007	Carl	20	male
s008	Cindy	21	female
s009	Haley	18	male
s010	Devin	20	male

图 4-86 数据表 student

下面分步骤讲解如何抽取数据表 student 的数据保存至 MySQL 数据库的数据表 student_mysql 中,具体步骤如下。

1. 打开 Kettle 工具,创建转换

使用 Kettle 工具创建转换 mssql_mysql,并添加“表输入”控件、“表输出”控件以及 Hop 跳连接线,具体效果如图 4-87 所示。

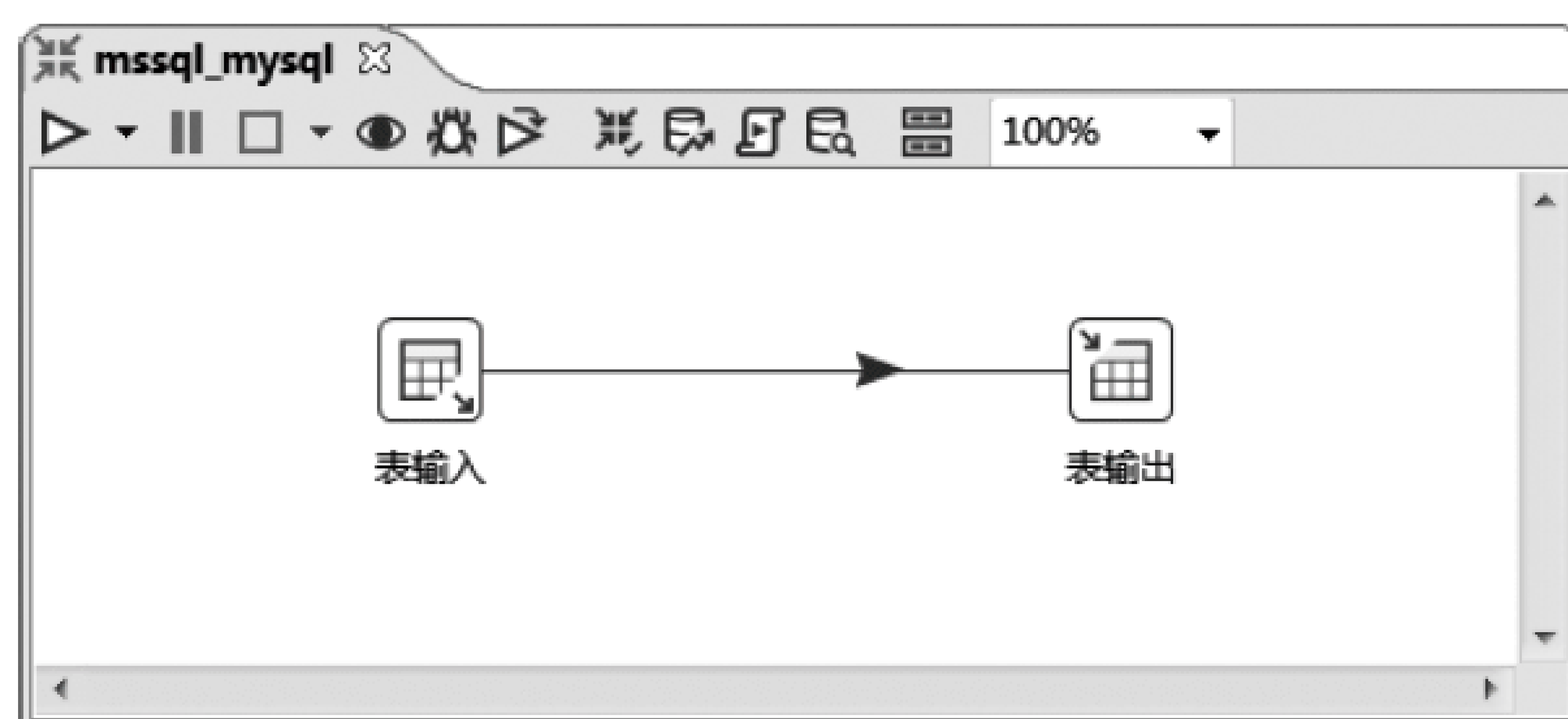


图 4-87 创建转换 mssql_mysql

2. 配置“表输入”控件

双击图 4-87 中的“表输入”控件,进入“表输入”界面,如图 4-88 所示。

在图 4-88 中单击“新建”按钮,配置数据库连接(注:配置数据库连接前需将 jtds-1.3.1.jar 驱动包添加至 Kettle 安装包的 lib 目录下),配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 4-89 所示。

在图 4-88 中的 SQL 框中编写查询数据表 student 的 SQL 语句,然后单击“预览”按钮,查看数据表 student 的数据是否成功从 SQL Server 数据库中抽取到表输入流中,具体如图 4-90 和图 4-91 所示。



图 4-88 “表输入”界面

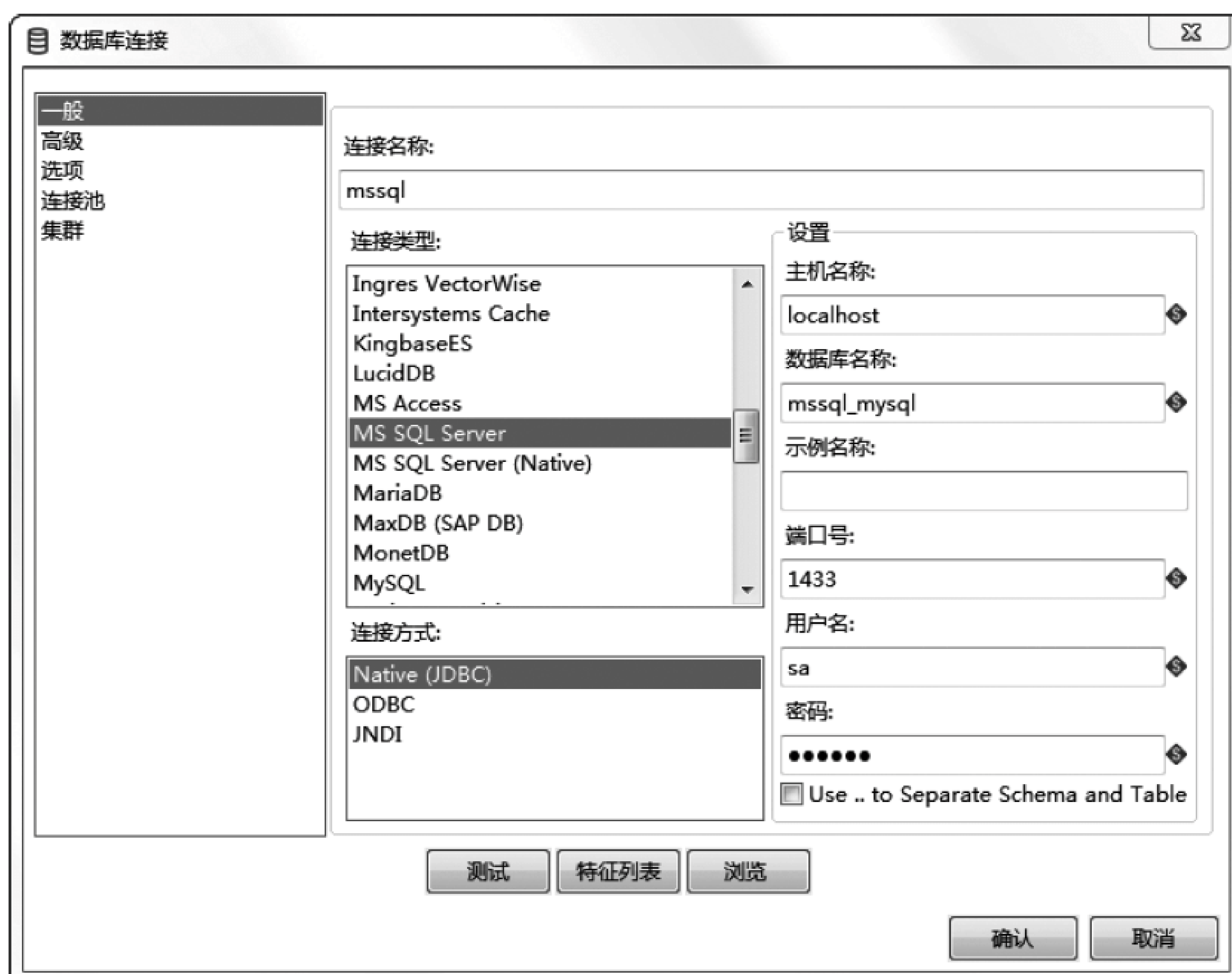


图 4-89 MySQL 数据库连接的配置

从图 4-91 中可以看出，数据表 student 的数据已经成功从 SQL Server 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“表输出”控件

双击图 4-87 中的“表输出”控件，进入“表输出”界面，如图 4-92 所示。

在图 4-92 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 4-93 所示。

单击图 4-92 中目标表后的“浏览”按钮，选择输出的目标表，即数据表 student（该表需



图 4-90　编写 SQL 语句

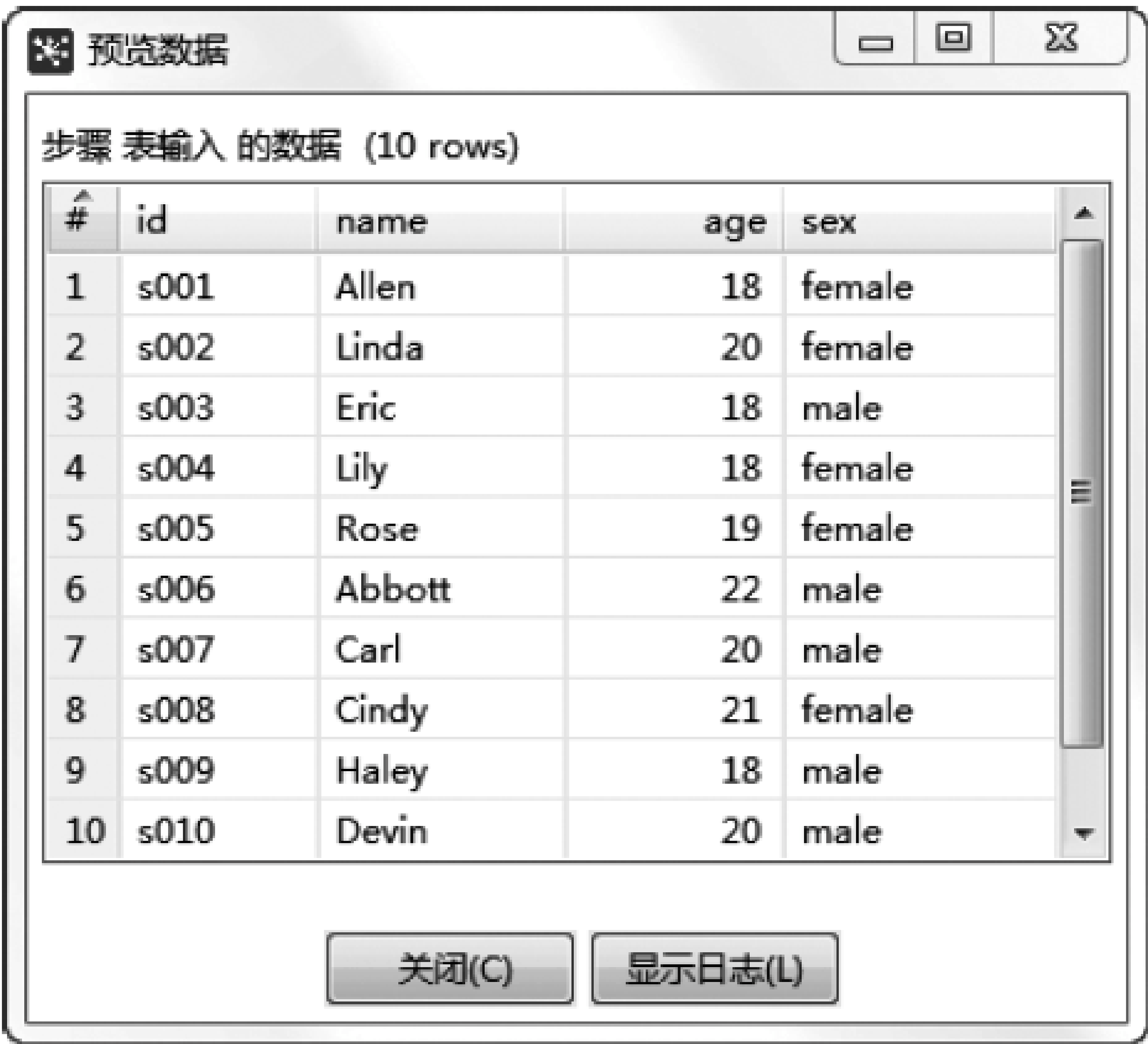


图 4-91　预览数据



图 4-92　“表输出”界面



图 4-93 MySQL 数据库连接的配置

提前创建，且表结构需根据 SQL Server 中的数据字段和数据类型创建，这里不演示）；勾选“指定数据库字段”复选框，用于将数据表 student 的字段与数据表 student_mysql 的字段进行匹配，具体如图 4-94 所示。



图 4-94 指定输出的目标表

在图 4-94 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，如图 4-95 所示。

在图 4-95 中依次选中“源字段”中的字段和“目标字段”中对应的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中。若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 4-96 所示。

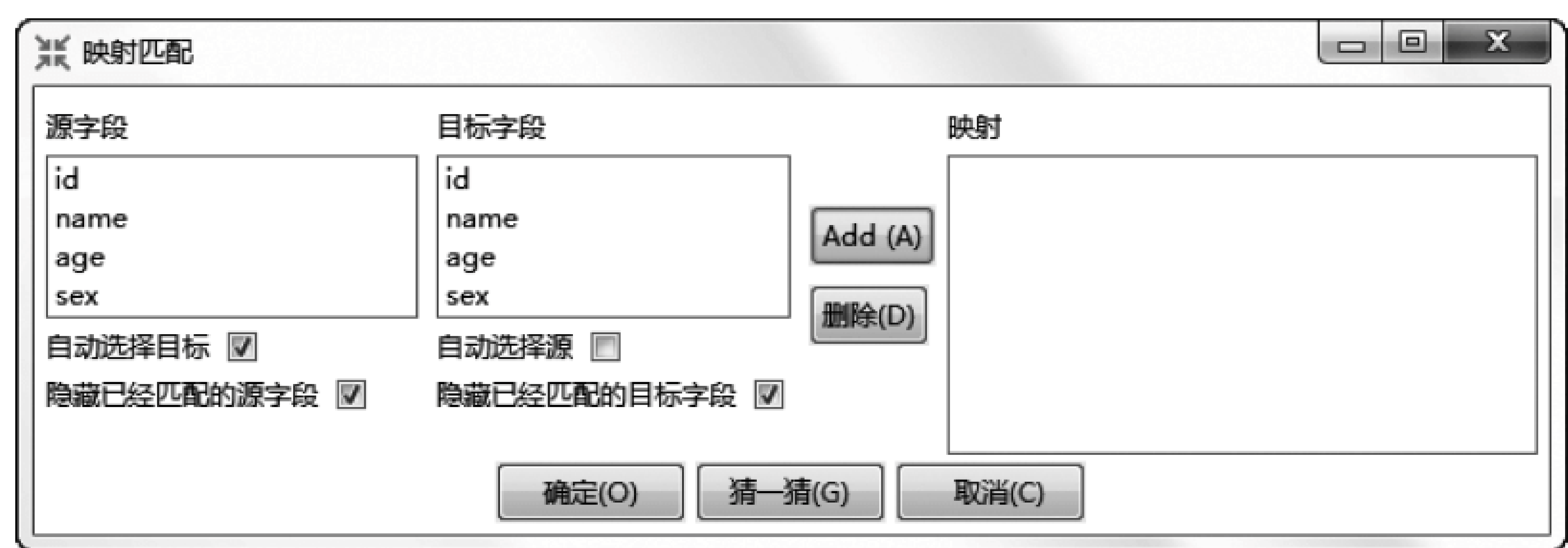


图 4-95　“映射匹配”对话框

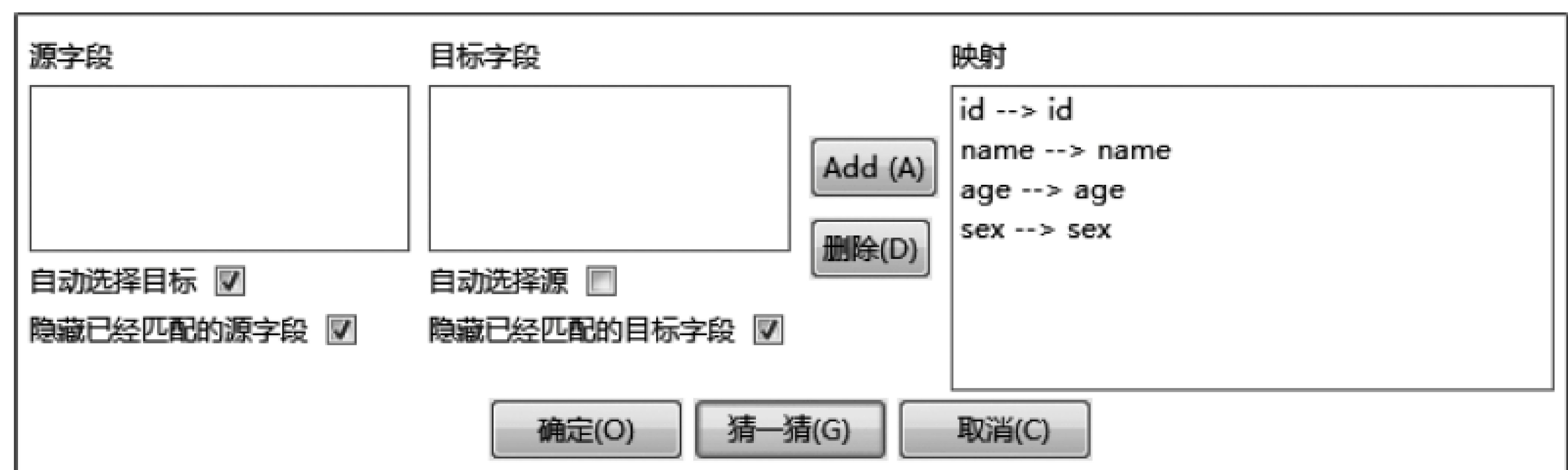


图 4-96　设置映射匹配

在图 4-96 中单击“映射匹配”对话框中的“确定”按钮，“表输出”界面最终显示的效果如图 4-97 所示，单击“确定”按钮，完成“表输出”控件的配置。

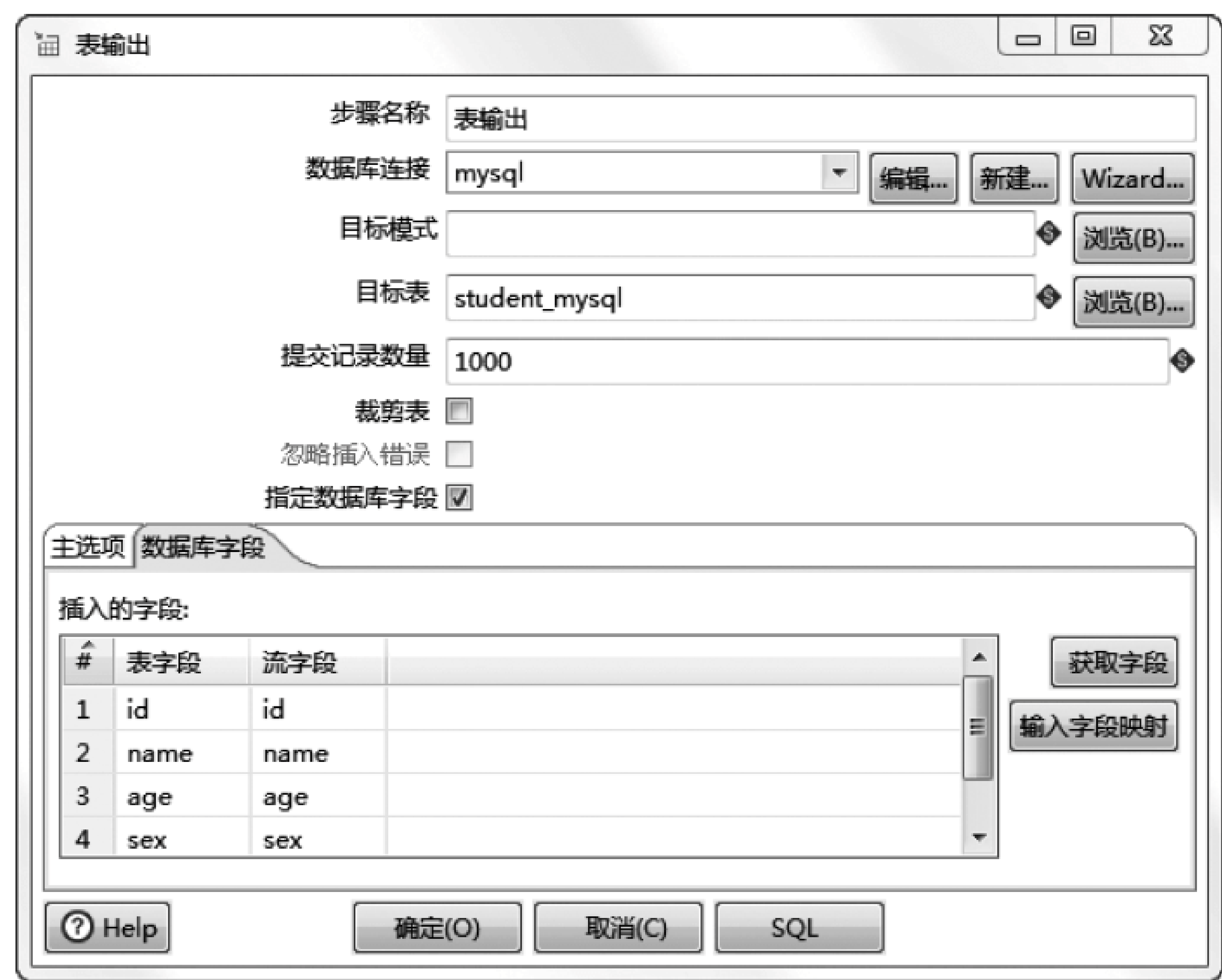


图 4-97　“表输出”界面最终显示的效果

4．运行转换 mssql_mysql

单击转换工作区顶部的▷按钮，运行创建的转换 mssql_mysql，实现将数据表 student

中的数据抽取到数据表 student_mysql 中，具体如图 4-98 所示。

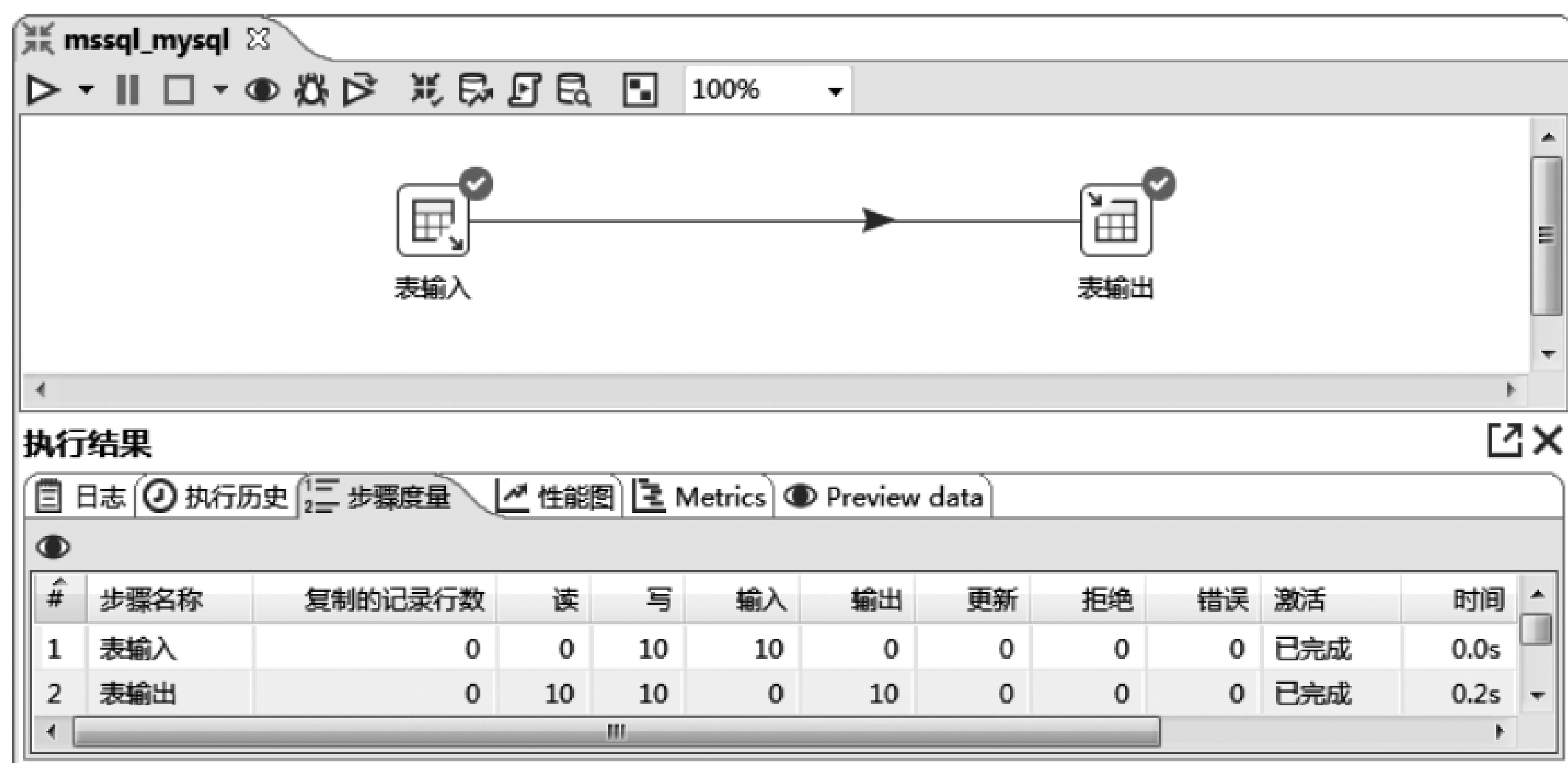


图 4-98　运行转换 mssql_mysql

从图 4-98 中执行结果的“步骤度量”可以看出，“表输入”控件输入 10 条数据并写入该控件中；“表输出”控件读取“表输入”控件的 10 条数据并写入该控件中，最终进行输出。也就是说，“表输出”控件将从表输入流中读取的 10 条数据均写入数据表 student_mysql 中。

5. 查看数据表 student_mysql 中的数据

通过 SQLyog 工具，查看数据表 student_mysql 是否已成功插入 10 条数据，查看结果如图 4-99 所示。

id	name	age	sex
s001	Allen	18	female
s002	Linda	20	female
s003	Eric	18	male
s004	Lily	18	female
s005	Rose	19	female
s006	Abbott	22	male
s007	Carl	20	male
s008	Cindy	21	female
s009	Haley	18	male
s010	Devin	20	male

数据库：extract　表格：student_mysql

图 4-99　数据表 student_mysql

从图 4-99 中可以看出，数据表 student_mysql 中已插入数据，说明我们成功实现了将 SQL Server 数据库的数据表 student 中的数据抽取到 MySQL 数据库的数据表 student_mysql 中。

4.3.2　抽取非关系型数据库的数据

看到 NoSQL 这个词，大家可能会误以为是 No! SQL 的缩写，并深感诧异：“SQL 怎么会没有必要了呢?”。实际上，NoSQL 是 Not Only SQL 的缩写，它的含义不仅是 SQL，为了弥补关系型数据库的不足，各种各样的 NoSQL 数据库应运而生，如 MongoDB、Redis 及

HBase 等非关系型数据库。

现有一个集合，名为 Personal information(存在于 MongoDB 数据库中，此集合需读者提前创建)，具体内容如图 4-100 所示。

_id	name	age	gender	hobby	address
ObjectID("5d09f7f40e36a41070ff08a7") ObjectID	"zhangsan" String	19 Int32	"male" String	"movie" String	"北京市昌平区和谐小区5号楼2单元001室" String
ObjectID("5d09f9540e36a41070ff08a9") ObjectID	"lisi" String	20 Int32	"female" String	"reading book" String	"上海市徐汇区文博苑小区6号楼1单元002室" String
ObjectID("5d09fa2b35d6671070c3ed8c") ObjectID	"wangwu" String	18 Int32	"male" String	"playing" String	"北京市朝阳区流星花园1号楼4单元003室" String
ObjectID("5d09fb8735d6671070c3ed8d") ObjectID	"xiaoqi" String	20 Int32	"female" String	"Listen to music" String	"北京市朝阳区文书苑3号楼1单元102室" String
ObjectID("5d09fc2935d6671070c3ed8e") ObjectID	"zhaoliu" String	19 Int32	"male" String	"Painting" String	"北京市海淀区中关村园11号楼1单元603室" String

图 4-100　Personal information 集合

下面分步骤讲解如何抽取集合 Personal information 中的数据并保存至 MySQL 中的数据表 Personal information 中，具体步骤如下。

1. 打开 Kettle 工具，创建转换

通过使用 Kettle 工具创建转换 mongodb_mysql_extract，并添加 MongoDB input 控件、JSON input 控件、“表输出”控件以及 Hop 跳连接线，具体效果如图 4-101 所示。

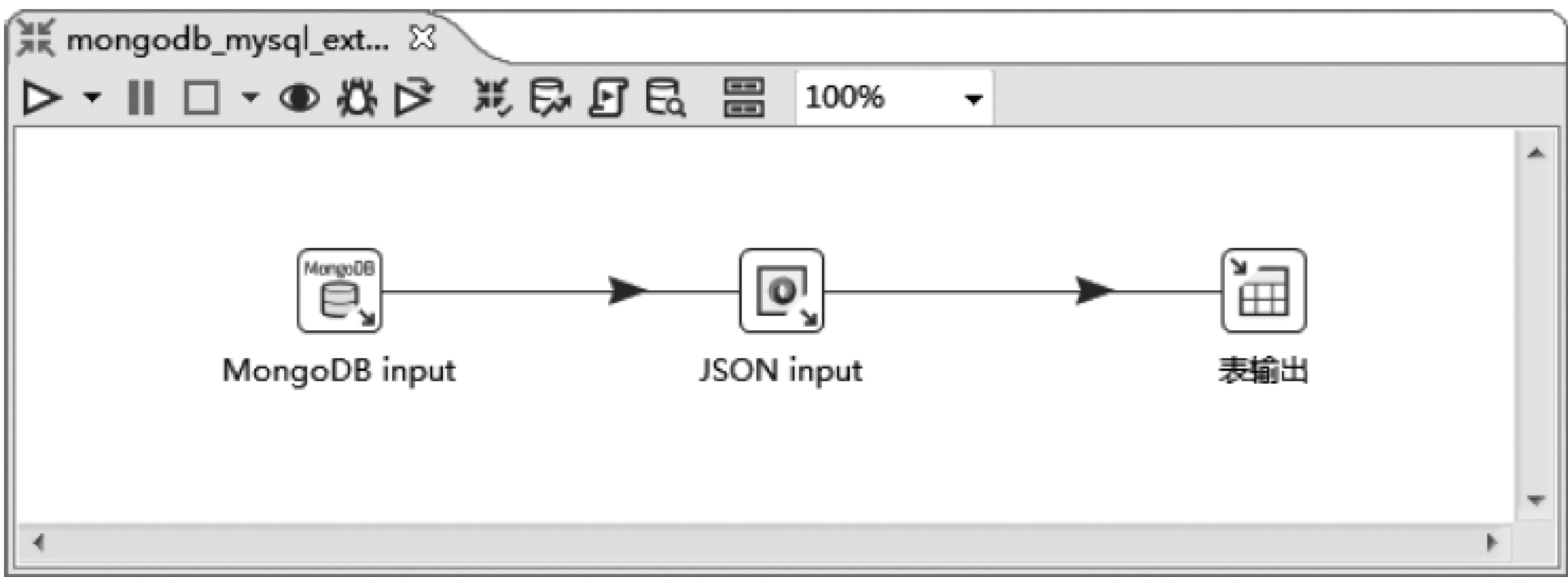


图 4-101　创建转换 mongodb_mysql_extract

2. 配置 MongoDB input 控件

双击图 4-101 中的 MongoDB input 控件，进入 MongoDB input 界面，如图 4-102 所示。

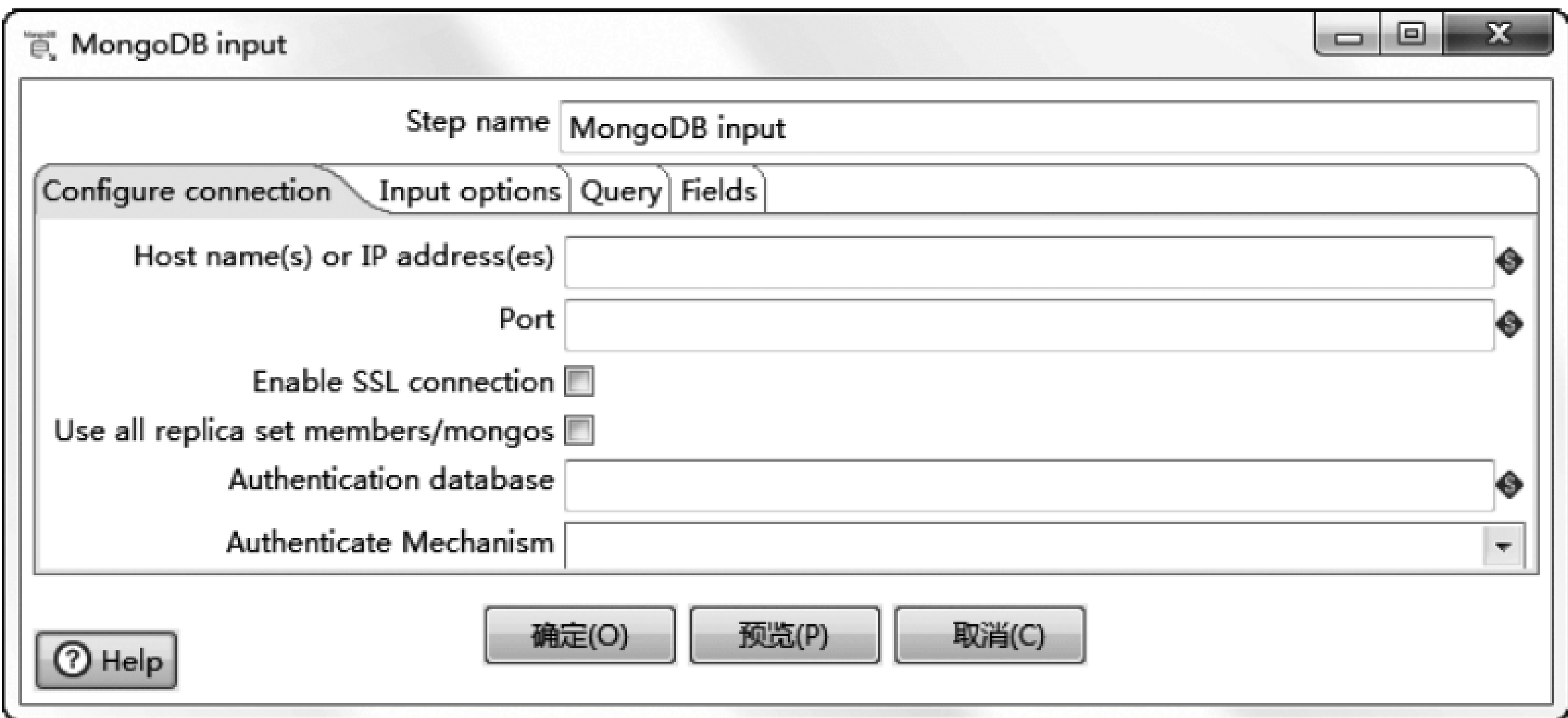


图 4-102　MongoDB input 界面

在图 4-102 中单击 Configure connection 选项卡，配置数据库连接，即在 Host name(s) or IP address(es)处指定主机名 localhost（本节使用的 MongoDB 数据库安装在本机），在 Port 处指定端口号 27017，具体效果如图 4-103 所示。

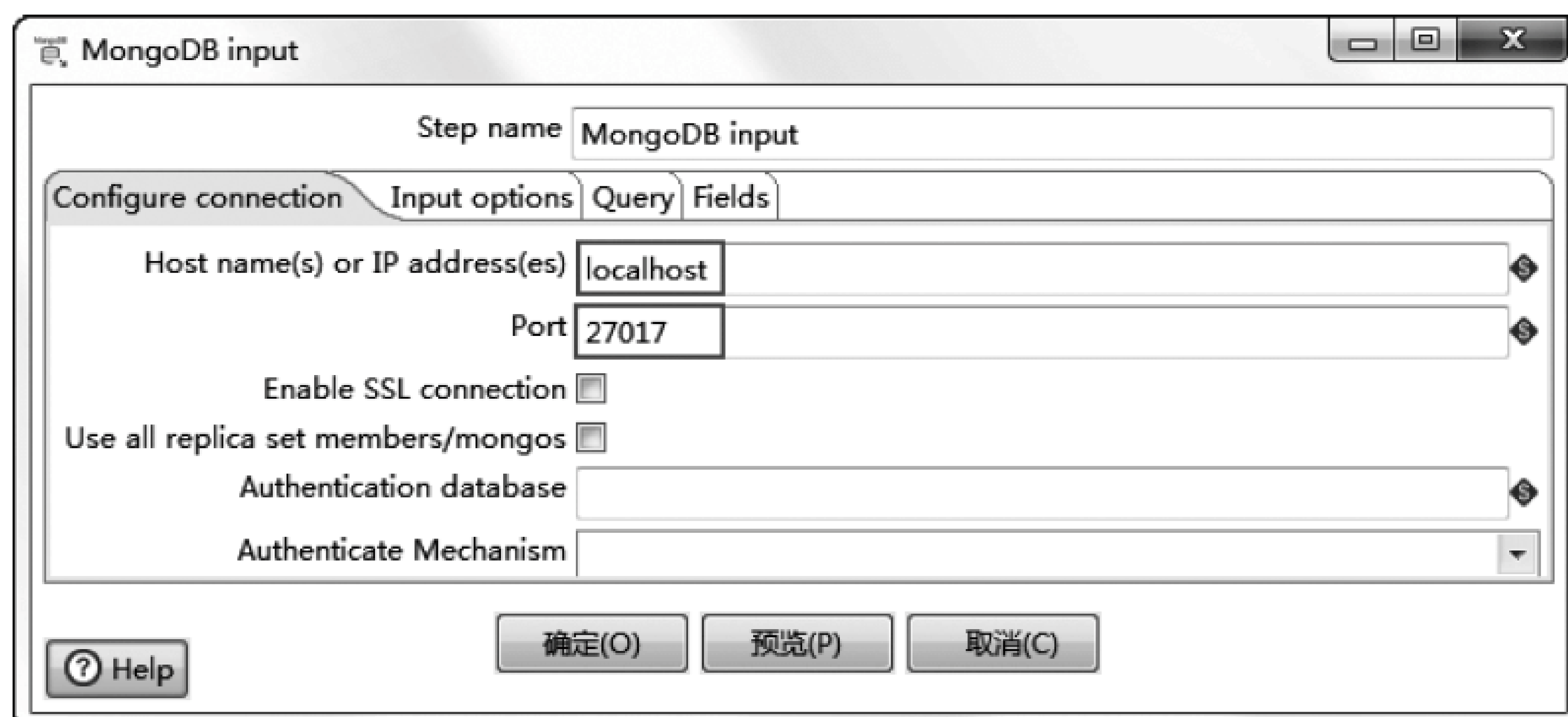


图 4-103 配置 MongoDB 数据库连接

在图 4-103 中单击 Input options 选项卡，进入 Input options 选项卡界面，如图 4-104 所示。

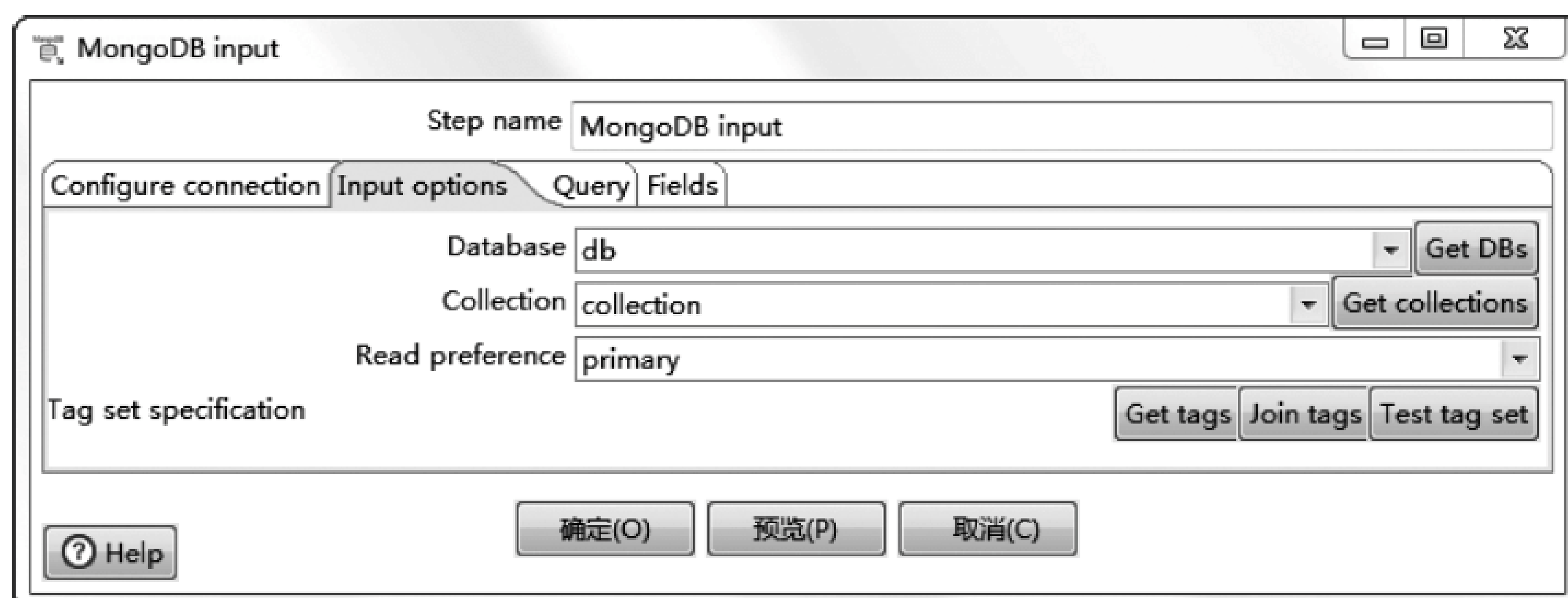


图 4-104 Input options 选项卡界面

在图 4-104 中指定数据库和数据表，即在 Database 处添加数据库 mongodb_mysql，在 Collection 处添加集合 Personal information，具体如图 4-105 所示。

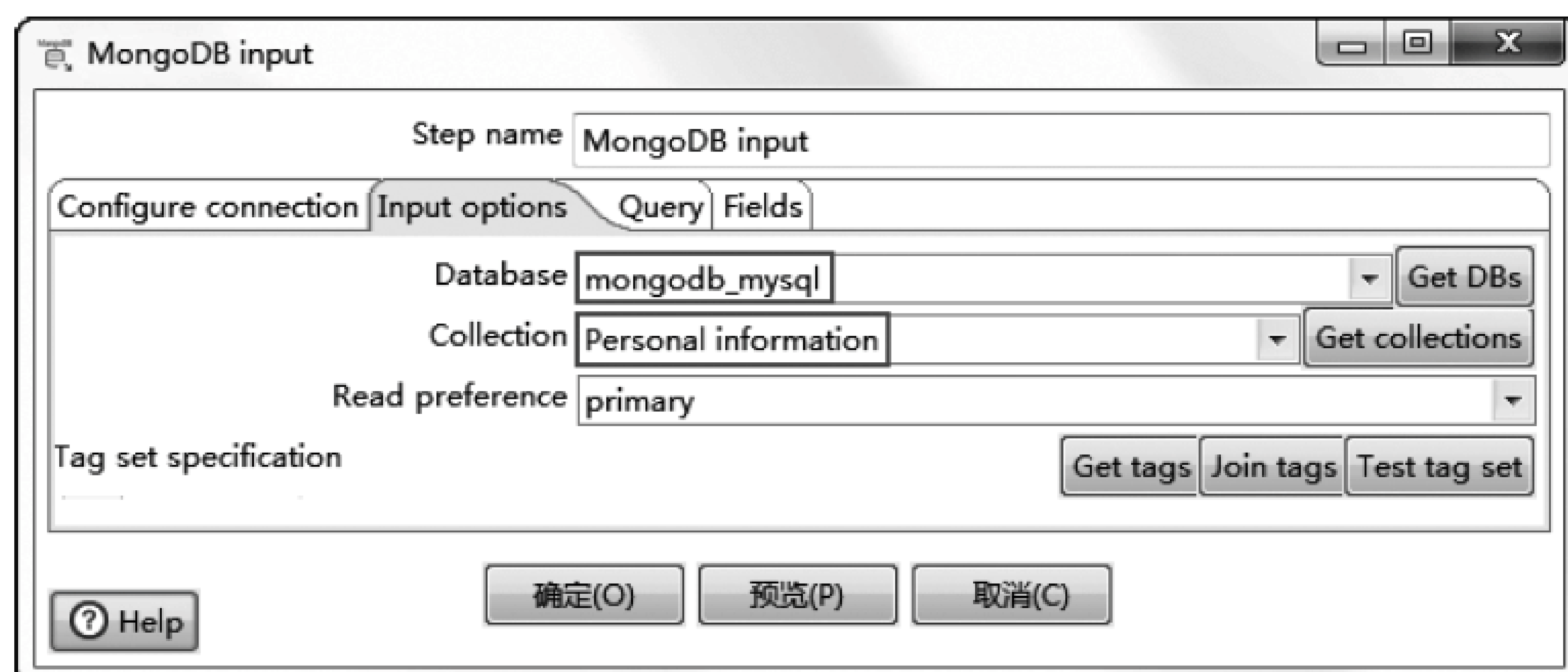


图 4-105 指定数据库和数据表

在图 4-105 中单击 Fields 选项卡，勾选 Output single JSON field 复选框，并在 Name of JSON output field 处指定输出的字段名为 json，具体如图 4-106 所示。

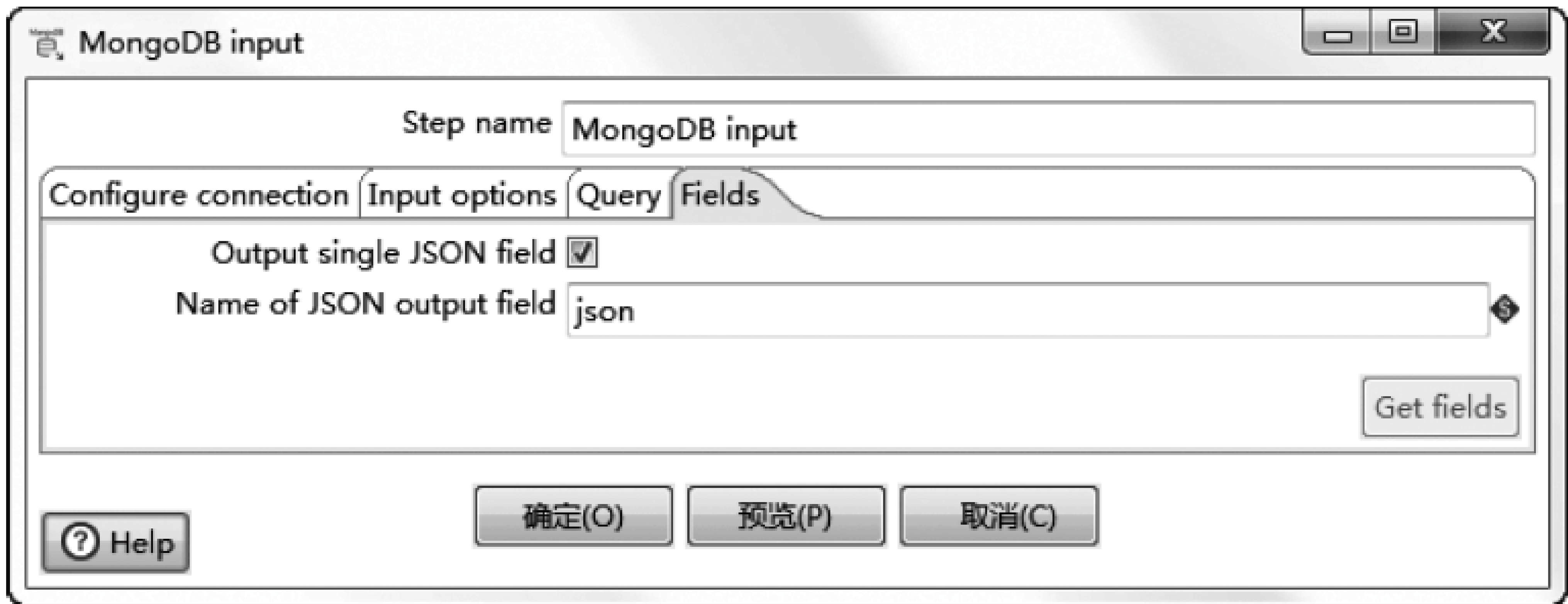


图 4-106　指定输出字段

在图 4-106 中单击"确定"按钮，完成 MongoDB input 控件的配置。

3．配置 JSON input 控件

双击图 4-101 中的 JSON input 控件，进入"JSON 输入"界面，如图 4-107 所示。



图 4-107　"JSON 输入"界面

在图 4-107 中单击"文件"选项卡，配置数据的获取源，勾选"源定义在一个字段里?"复选框，在"从字段获取源"后的下拉列表中选择 json，具体效果如图 4-108 所示。



图 4-108　配置数据的获取源

在图 4-108 中单击“字段”选项卡，添加要抽取的数据字段，具体如图 4-109 所示。



图 4-109 添加要抽取的字段

在图 4-109 中单击“确定”按钮，完成 JSON input 控件的配置。

4. 配置表输出控件

双击图 4-101 中的“表输出”控件，进入“表输出”界面，如图 4-110 所示。



图 4-110 “表输出”界面

在图 4-110 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 4-111 所示。

单击图 4-110 中目标表右侧的“浏览”按钮，选择输出的目标表，即数据表 personal information（该表需提前创建，且表字段需根据 MongoDB 数据库的集合 Personal information 中的数据字段和数据类型创建，这里不演示）；勾选“指定数据库字段”复选框，用于将 MySQL 中数据表 Personal information 的字段与 MongoDB 中集合 Personal information 的字段进行匹配，具体如图 4-112 所示。

在图 4-112 中单击“数据库字段”选项卡，进入“数据库字段”选项卡界面，如图 4-113



图 4-111　MySQL 数据库连接的配置



图 4-112　指定输出的目标表

所示。

在图 4-113 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，如图 4-114 所示。

在图 4-114 中依次选中“源字段”中的字段和“目标字段”中对应的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 4-115 所示。

在图 4-115 中单击“映射匹配”对话框中的“确定”按钮，“表输出”界面最终显示的效果如图 4-116 所示，单击“确定”按钮，完成“表输出”控件的配置。



图 4-113 “数据库字段”选项卡界面

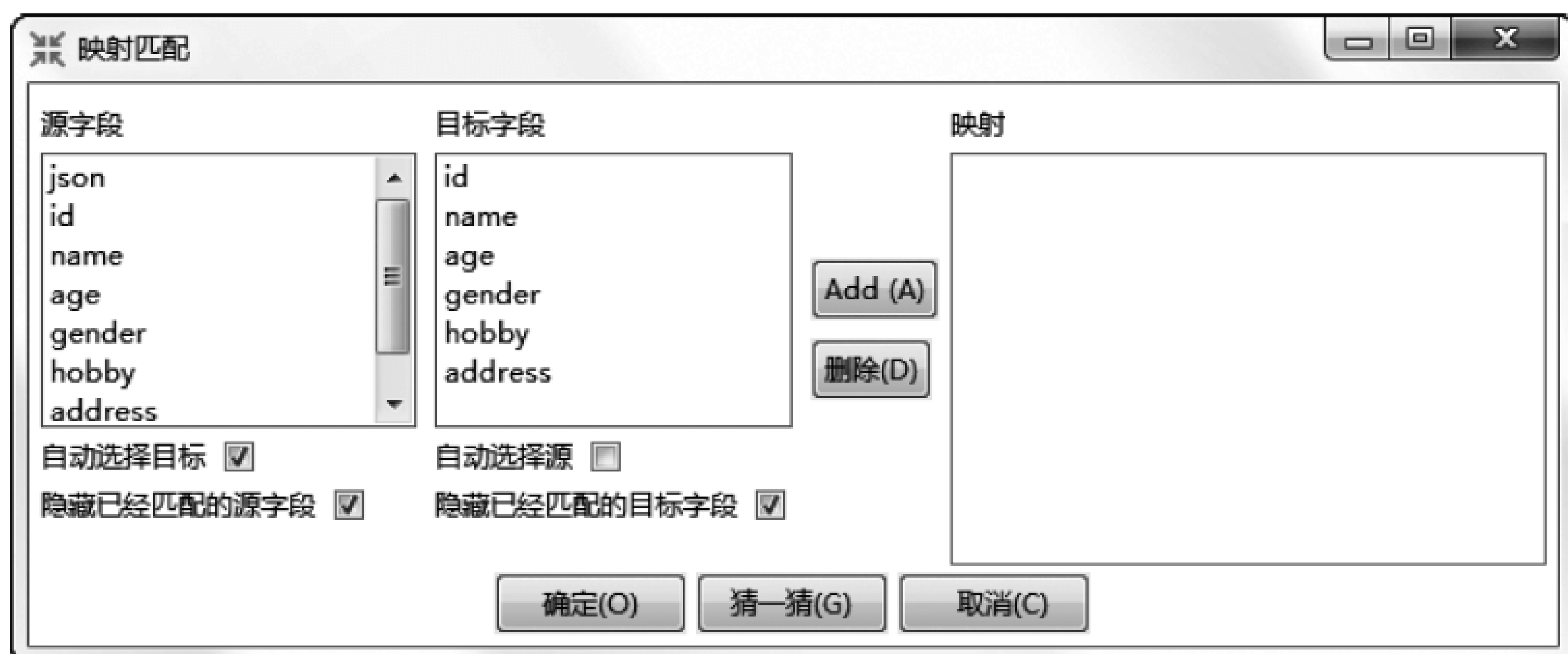


图 4-114 “映射匹配”对话框



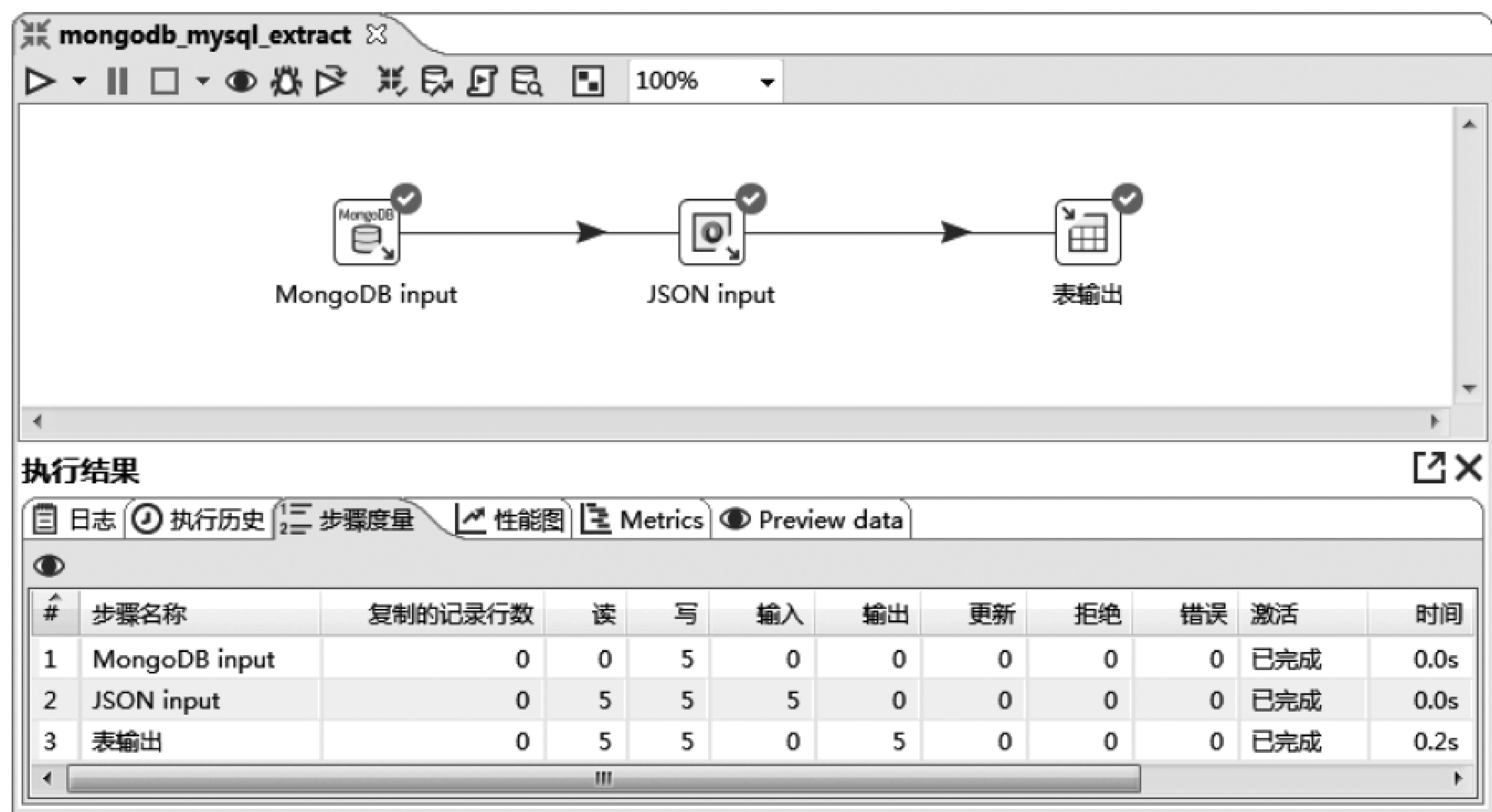
图 4-115 设置映射匹配

5. 运行转换 mongodb_mysql_extract

单击转换工作区顶部的▷按钮，运行创建的转换 mongodb_mysql_extract，实现将集合 Personal information 中的数据抽取到 MySQL 的数据表 Personal information 中，具体如图 4-117 所示。



图 4-116　“表输出”界面最终显示的效果



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活	时间
1	MongoDB input	0	0	5	0	0	0	0	0	已完成	0.0s
2	JSON input	0	5	5	5	0	0	0	0	已完成	0.0s
3	表输出	0	5	5	0	5	0	0	0	已完成	0.2s

图 4-117　运行转换 mongodb_mysql_extract

从图 4-117 中执行结果的“步骤度量”可以看出，MongoDB input 控件写入 5 条数据；JSON input 控件读取 MongoDB input 控件的 5 条数据作为输入并写入该控件中；“表输出”控件读取 JSON input 控件的 5 条数据并写入该控件，最终进行输出。也就是说，“表输出”控件将从 JSON input 输入流中读取的 5 条数据写入数据表 Personal information 中。

6. 查看数据表 Personal information 中的数据

通过 SQLyog 工具，查看数据表 Personal information 是否已成功插入 5 条数据，查看

结果如图 4-118 所示。

id	name	age	gender	hobby	address
5d09f7f40e36a41070ff08a7	zhangsan	19	male	movie	北京市昌平区和谐小区5号楼2单元001室
5d09f9540e36a41070ff08a9	lisi	20	female	reading book	上海市徐汇区文博苑小区6号楼1单元002室
5d09fa2b35d6671070c3ed8c	wangwu	18	male	playing	北京市朝阳区流星花园1号楼4单元003室
5d09fb8735d6671070c3ed8d	xiaoqi	20	female	Listen to music	北京市朝阳区文书苑3号楼1单元102室
5d09fc2935d6671070c3ed8e	zhaoliu	19	male	Painting	北京市海淀区中关村园11号楼1单元603室

图 4-118 数据表 Personal information

从图 4-118 中可以看出，数据表 Personal information 中已插入数据，说明我们成功实现了将 MongoDB 的集合 Personal information 中的数据抽取到 MySQL 的数据表 Personal information 中。

4.4 本章小结

本章主要讲解了数据抽取的相关知识，包括抽取文本数据、抽取 Web 数据以及抽取数据库数据。希望读者通过本章的学习，掌握抽取各种形式的数据保存至数据库中，以便于后续对数据进行清洗和分析。

4.5 本章习题

一、填空题

1. 实际应用中，常用的文本文件类型有两种，分别是 TSV 文件和________文件。
2. 制表符文件中的数据是以________的结构进行存储。
3. 使用________分隔数据字段的文件被称为逗号分隔值文件。
4. HTML 可以以________的形式展示，HTML 文档中包含________和纯文本。
5. ________是一种轻量级的数据交换格式。

二、判断题

1. XML 是一种和 HTML 完全相同的标记语言。 ( )
2. JSON 是一种编程语言。 ( )
3. 通过制表符分隔的文本数据与未使用制表符分隔的数据相比，前者更便于观察识别，同时也便于对数据进行抽取操作。 ( )
4. CSV 文件以纯文本形式存储表格数据(数字和文本)。 ( )
5. NoSQL 是 No! SQL 的缩写。 ( )

三、选择题

1. 下列实现原则中，选项________不属于 CSV 的实现原则。

   A. 文件开头不能留空，以“列”为单位

   B. 在文件读写时，引号和逗号操作规则不可互逆

C. 文件中不支持数字或特殊字符

D. 文件中的一行数据不能跨行,但是行与行间可存在空行

2. 下列关于 XML 用途的说法中,________的说法是正确的。

A. XML 不可将数据从 HTML 中分离

B. XML 无法简化数据共享

C. XML 无法使数据充分利用

D. XML 可用于创建新的互联网语言

3. 下列数据库中,数据库________属于非关系型数据库。

A. MySQL B. MongoDB C. Oracle D. SQL Server

四、操作题

通过 Kettle 工具,实现以下功能:

(1) 抽取 CSV 文件 csv_extract.csv 中的数据保存至数据库 extract 中的数据表 csv 中。

(2) 抽取 JSON 文件 json_extract.js 中的数据保存至数据库 extract 中的数据表 json 中。

第 5 章 数据的清洗与检验

学习目标

(1) 掌握数据去重的方法

(2) 掌握缺失值处理的方法

(3) 掌握异常值处理的方法

(4) 了解数据检验的作用

数据清洗是一项复杂且烦琐的工作，同时也是整个数据分析过程中最重要的环节。数据清洗的目的在于提高数据质量，将"脏"数据("脏"数据在这里指的是对数据分析没有实际意义、格式非法、不在指定范围内的数据)清洗干净，使原数据具有完整性、唯一性、权威性、合法性、一致性等特点。常见的数据清洗操作包括重复值的处理、缺失值的处理、异常值的处理等操作，同时，为了保证数据的有效性，少不了数据校验操作。本章将针对数据清洗和校验进行详细讲解。

5.1 数据去重

数据去重又称重复数据的删除，通常指的是找出数据文件集合中重复的数据并将其删除，只保存唯一的数据单元，从而消除冗余数据。通常，数据去重方法分为两种，分别是完全去重和不完全去重。本节将针对这两种数据去重的方式进行讲解。

5.1.1 完全去重

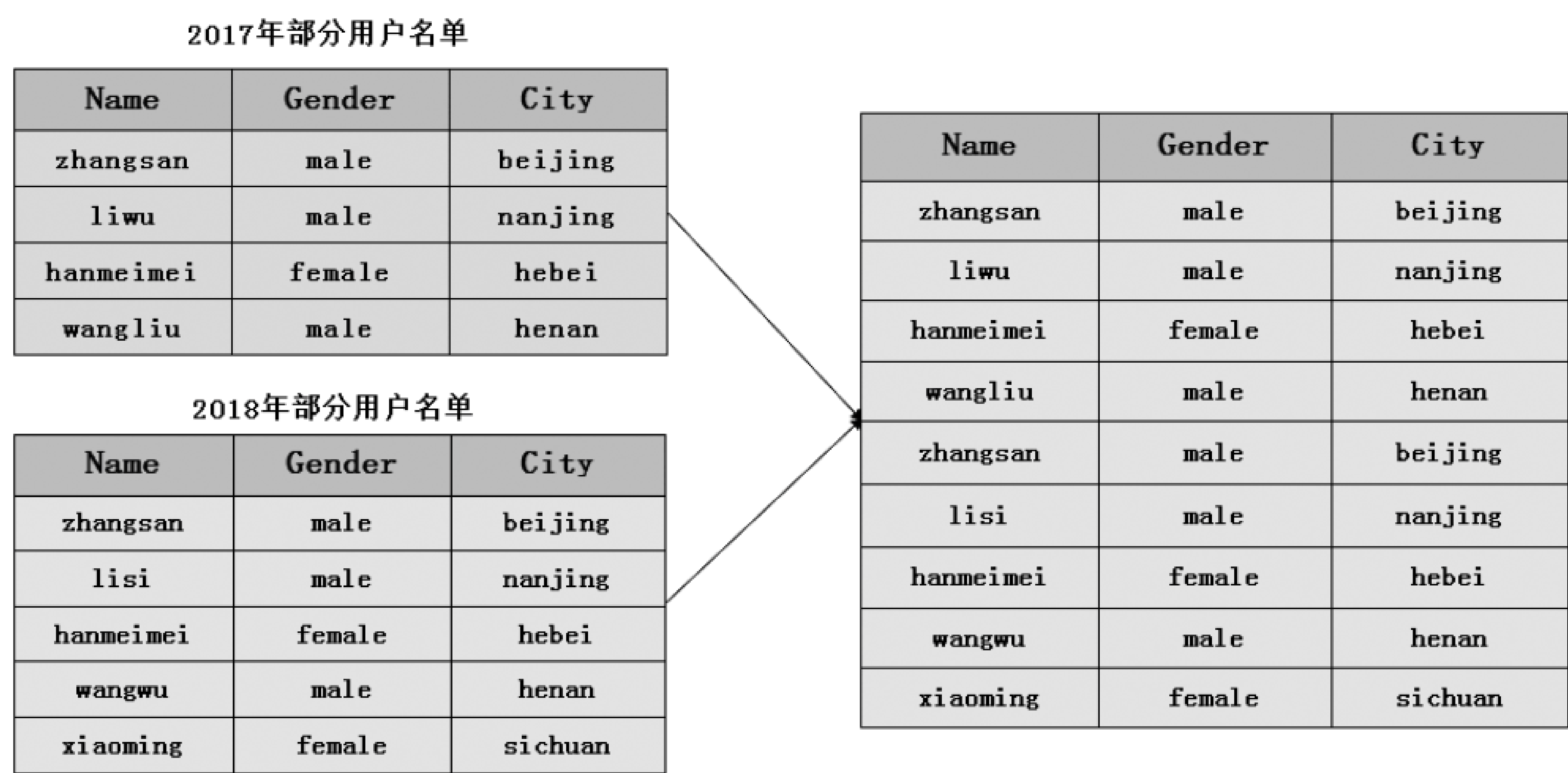
完全去重指的是消除完全重复的数据，这里提到的完全重复数据指的是数据表记录字段值完全一样的数据。例如，现在有两个表格，分别记录不同年份的用户信息，现要求合并统计所有用户信息，发现合并后的表格存在完全重复的数据，示例如图 5-1 所示。

从图 5-1 中可以看出，合并后的数据表中存在完全相同的数据，为了便于后期更加方便地使用这些用户数据，通常情况下会对数据进行去重操作。

接下来使用 Kettle 工具演示如何消除这些完全重复的数据，具体步骤如下。

1. 数据准备

合并后的用户名单存放在 CSV 文件 merge.csv 中，具体内容如图 5-2 所示。



2017年部分用户名单

Name	Gender	City
zhangsan	male	beijing
liwu	male	nanjing
hanmeimei	female	hebei
wangliu	male	henan

2018年部分用户名单

Name	Gender	City
zhangsan	male	beijing
lisi	male	nanjing
hanmeimei	female	hebei
wangwu	male	henan
xiaoming	female	sichuan

Name	Gender	City
zhangsan	male	beijing
liwu	male	nanjing
hanmeimei	female	hebei
wangliu	male	henan
zhangsan	male	beijing
lisi	male	nanjing
hanmeimei	female	hebei
wangwu	male	henan
xiaoming	female	sichuan

图 5-1　用户名单

	A	B	C
1	Name	Gender	City
2	zhangsan	male	beijing
3	liwu	male	nanjing
4	hanmeimei	female	hebei
5	wangliu	male	henan
6	zhangsan	male	beijing
7	lisi	male	nanjing
8	hanmeimei	female	hebei
9	wangwu	male	henan
10	xiaoming	female	sichuan

图 5-2　merge.csv 文件的内容

2. 打开 Kettle 工具，新建转换

使用 Kettle 工具创建转换 repeat_transform，并添加“CSV 文件输入”控件、“唯一行(哈希值)”控件以及 Hop 跳连接线，具体效果如图 5-3 所示。

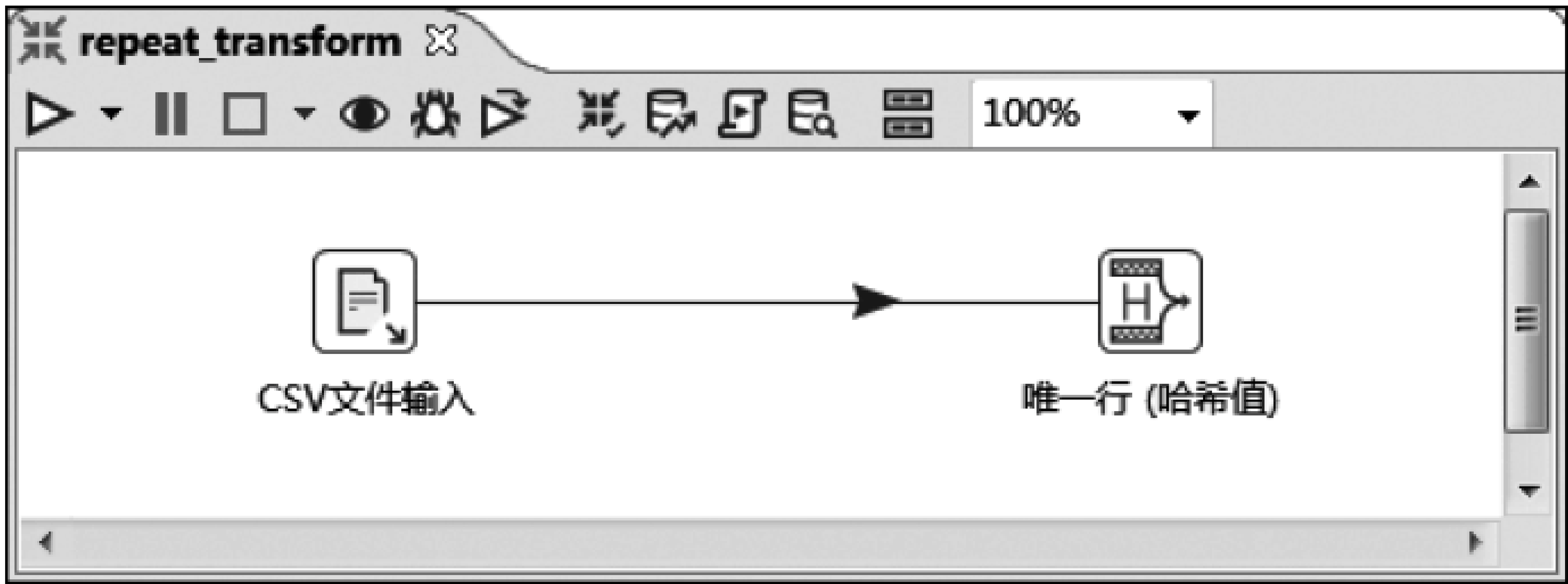


图 5-3　创建转换 repeat_transform

3. 配置“CSV 文件输入”控件

双击图 5-3 中的“CSV 文件输入”控件，进入“CSV 文件输入”界面，如图 5-4 所示。



图 5-4 “CSV 文件输入”界面

在图 5-4 中单击“浏览”按钮，选择要进行完全去重处理的 CSV 文件 merge.csv，如图 5-5 所示。



图 5-5 选择要进行完全去重处理的 CSV 文件 merge.csv

在图 5-5 中单击“获取字段”按钮，Kettle 会自动检索 CSV 文件，并对文件中的字段类型、格式、长度、精度等属性进行分析，具体如图 5-6 所示。

在图 5-6 中单击“预览”按钮，查看 CSV 文件 merge.csv 的数据是否加载到 CSV 文件输入流中，具体效果如图 5-7 所示。



图 5-6　Kettle 检索 CSV 文件



图 5-7　预览数据

从图 5-7 中可以看出，CSV 文件 merge.csv 的数据已经成功抽取到 CSV 文件输入流中，单击“关闭”→“确定”按钮，完成“CSV 文件输入”控件的配置。

4. 配置“唯一行(哈希值)”控件

双击图 5-3 中的“唯一行(哈希值)”控件，进入“唯一行(哈希值)”界面，如图 5-8 所示。

在图 5-8 中的“用来比较的字段”处添加要去重的字段，这里可以单击“获取”按钮，添加需要去重的字段，具体如图 5-9 所示。

在图 5-9 中单击“确定”按钮，完成“唯一行(哈希值)”控件的配置。

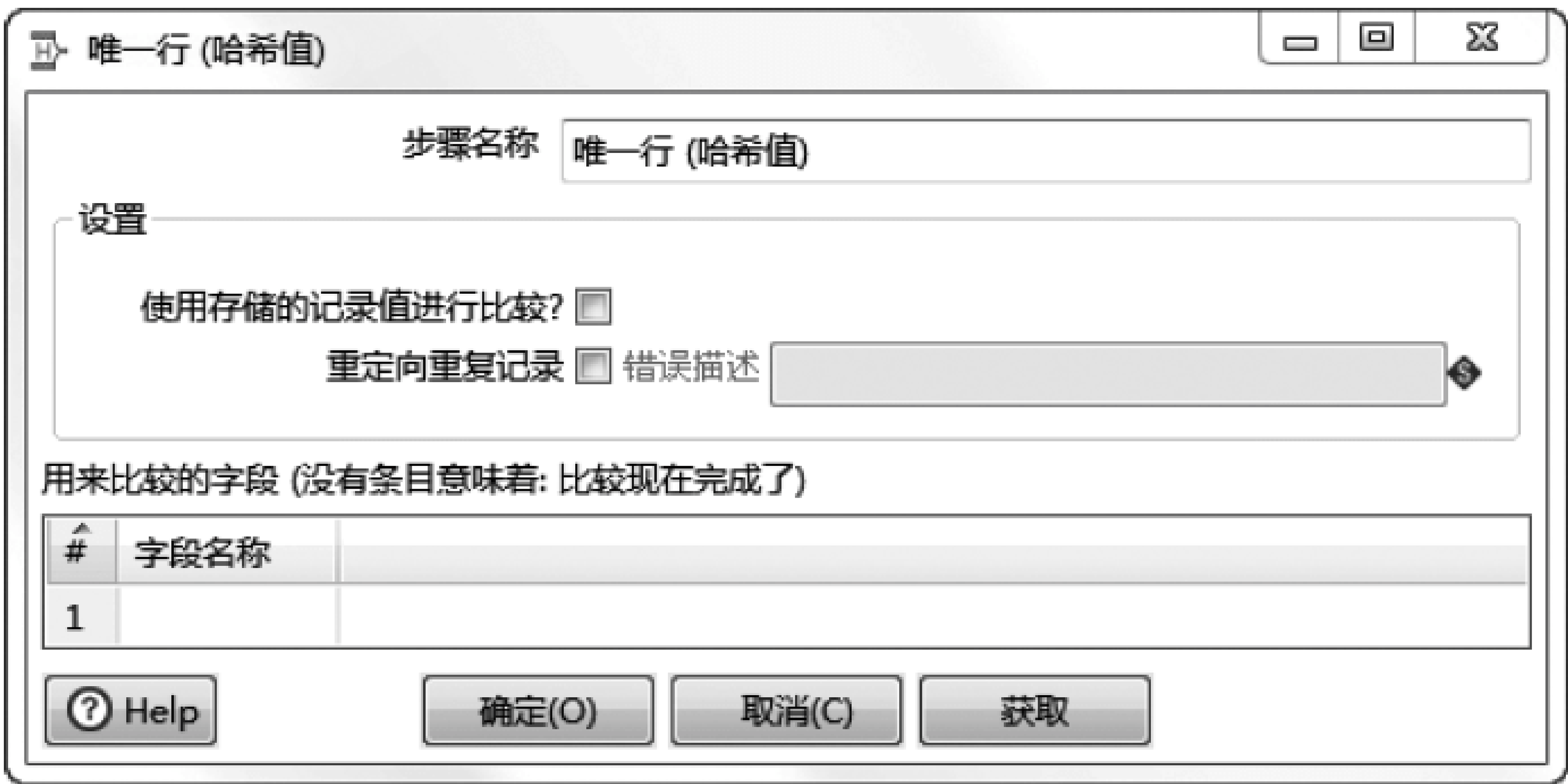


图 5-8　“唯一行(哈希值)”界面



图 5-9　添加需要去重的字段

5. 运行转换 repeat_transform

单击转换工作区顶部的▷按钮,运行创建的转换 repeat_transform,实现消除 CSV 文件 merge. csv 中完全重复的数据,具体如图 5-10 所示。

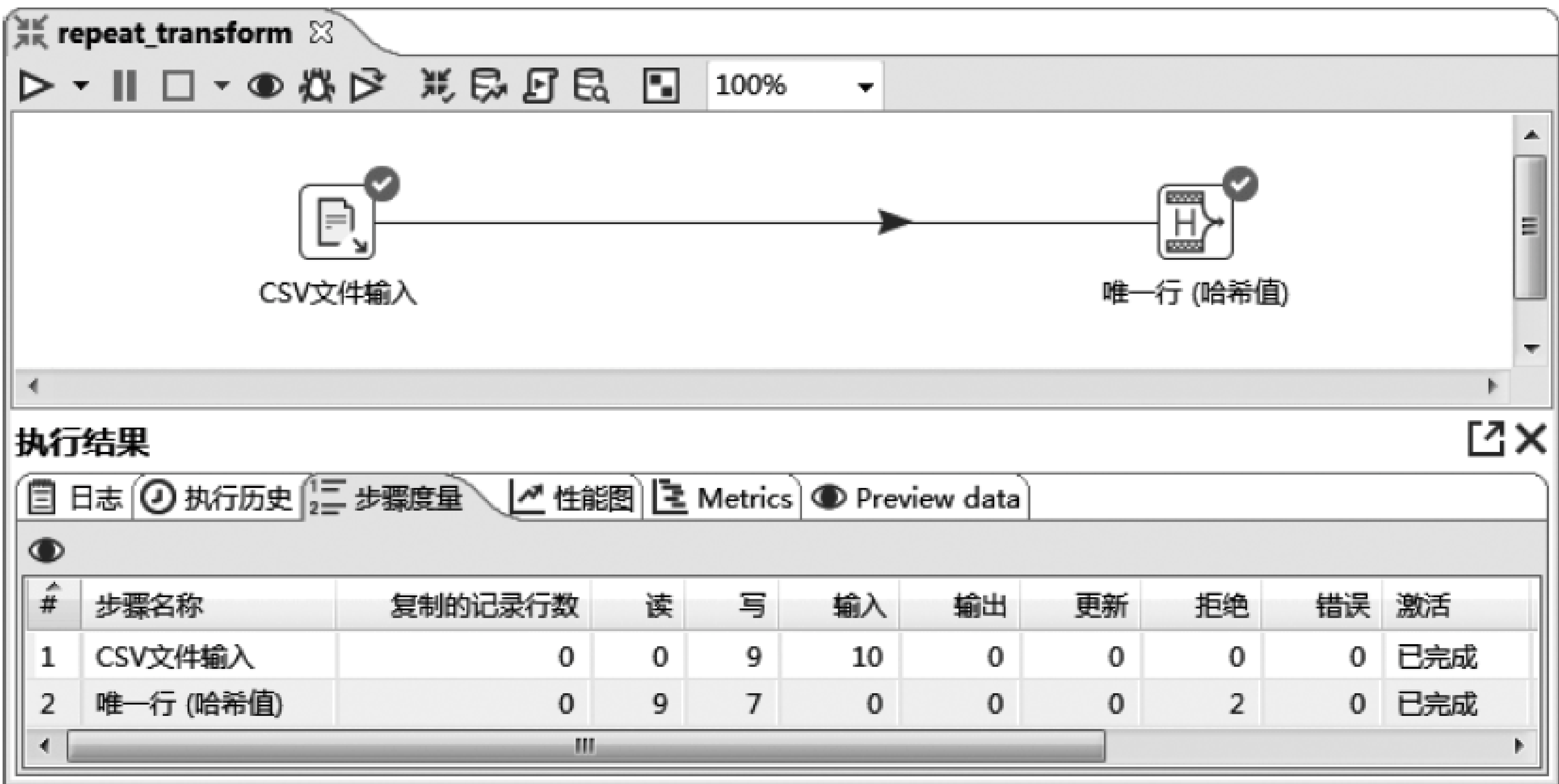


图 5-10　运行转换 repeat_transform

从图5-10中执行结果窗口的“步骤度量”选项卡可以看出,“CSV文件输入”控件输入10条数据并写入该控件9条数据(其中有1条数据为表头);“唯一行(哈希值)”控件从“CSV文件输入”控件读取9条数据,写入该控件7条数据,拒绝2条数据(这2条数据为完全重复数据)。也就是说,CSV文件merge.csv中有2条数据与其他数据完成重复。

选中图5-10中的“唯一行(哈希值)”控件,单击执行结果窗口的Preview data选项卡,查看是否消除CSV文件merge.csv中完全重复的数据,具体如图5-11所示。

执行结果

日志 | 执行历史 | 步骤度量 | 性能图 | Metrics | Preview data

${TransPreview.FirstRows.Label} ${TransPreview.LastRows.Label} ${TransPre

#	Name	Gender	City
1	zhangsan	male	beijing
2	liwu	male	nanjing
3	hanmeimei	female	hebei
4	wangliu	male	henan
5	lisi	male	nanjing
6	wangwu	male	henan
7	xiaoming	female	sichuan

图5-11 查看是否消除CSV文件merge.csv中完全重复的数据

从图5-11中可以看出,CSV文件merge.csv中的数据已经没有了完全重复的值,说明通过Kettle工具实现了消除完全重复数据的功能。

注意:本小节操作只是将CSV文件merge.csv的数据读取到Kettle中进行完全去重处理,并不会改变CSV文件merge.csv的原始数据,如需保存处理后的数据,须添加相关输出控件。

5.1.2 不完全去重

数据清洗过程中,所有字段值都相等的重复值是一定要剔除的。根据不同的业务场景,有时还需要选取其中若干字段进行去重操作。

假设现在有一份用户访问网站的数据文件people.txt,具体内容如图5-12所示。

```
garcia  MemberUser   15201011506 2019-03-17 11:59:12
anderson    GeneralUser 13105916651 2019-04-02 12:00:32
sharma  MemberUser   15803924337 2019-04-12 09:45:33
smith   GeneralUser 15905741651 2019-04-22 09:55:44
johnson MemberUser   15804058842 2019-05-23 09:12:23
garcia  GeneralUser 13404840224 2019-05-17 08:03:12
anderson    GeneralUser 13202438054 2019-05-12 12:17:53
jones   MemberUser   13606051520 2019-05-22 10:21:03
johnson GeneralUser 15000220514 2019-11-16 12:56:47
jones   MemberUser   13606051520 2019-11-02 13:01:35
garcia  MemberUser   15201011506 2019-08-24 15:17:57
```

图5-12 文件people.txt的内容

从图5-12中可以看出,第1条记录和第11条记录都指向同一个用户,只是访问的时间不同。下面通过Kettle工具将文件people.txt中不完全重复的数据进行去重处理,即使用Name(姓名)、UserLevel(用户级别)和Phone(手机号)3个字段作为去重处理的比较对象判

断唯一用户，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 part_repeat_transform，并添加“文本文件输入”控件、“唯一行(哈希值)”控件以及 Hop 跳连接线，具体效果如图 5-13 所示。



图 5-13 创建转换 part_repeat_transform

2. 配置“文本文件输入”控件

双击图 5-13 中的“文本文件输入”控件，进入“文本文件输入”界面，如图 5-14 所示。



图 5-14 “文本文件输入”界面

在图 5-14 中单击“浏览”按钮，选择要去重的文件 people. txt，效果如图 5-15 所示。

在图 5-15 中单击“增加”按钮，将要去重的文件 people. txt 添加到转换 part_repeat_transform 中，具体效果如图 5-16 所示。

在图 5-16 中单击“内容”选项卡，切换到“内容”选项卡界面，如图 5-17 所示。

在图 5-17 中清除分隔符处的默认分隔符“;”，并单击 Insert TAB 按钮，在分隔符处插入一个制表符；取消勾选“头部”复选框，若不取消，则在进行数据抽取操作时会排除文件第一行的数据。“内容”选项卡的配置如图 5-18 所示。

在图 5-18 中单击“字段”选项卡，切换到“字段”选项卡界面，如图 5-19 所示。

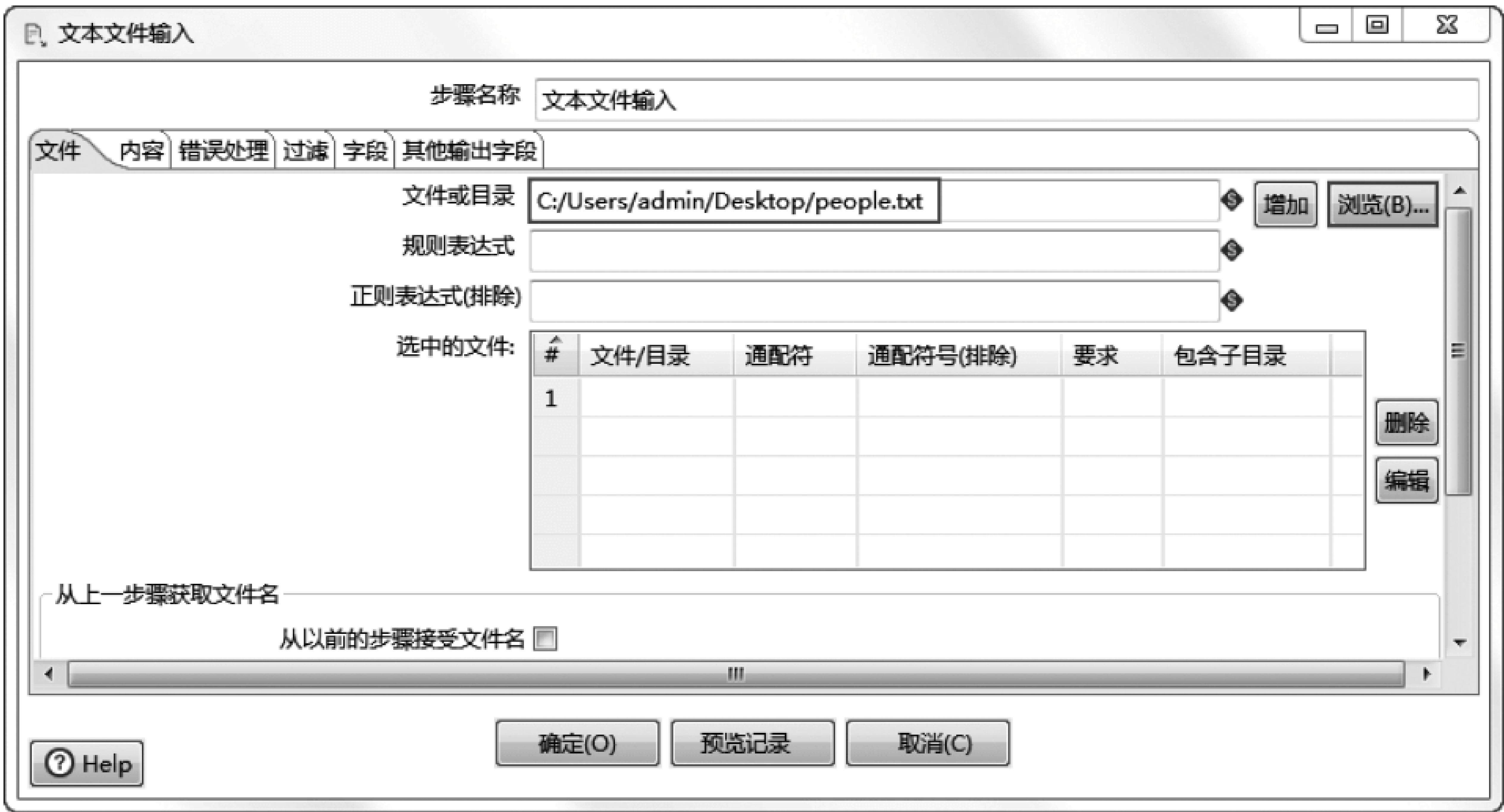


图 5-15　选择要去重的文件 people.txt

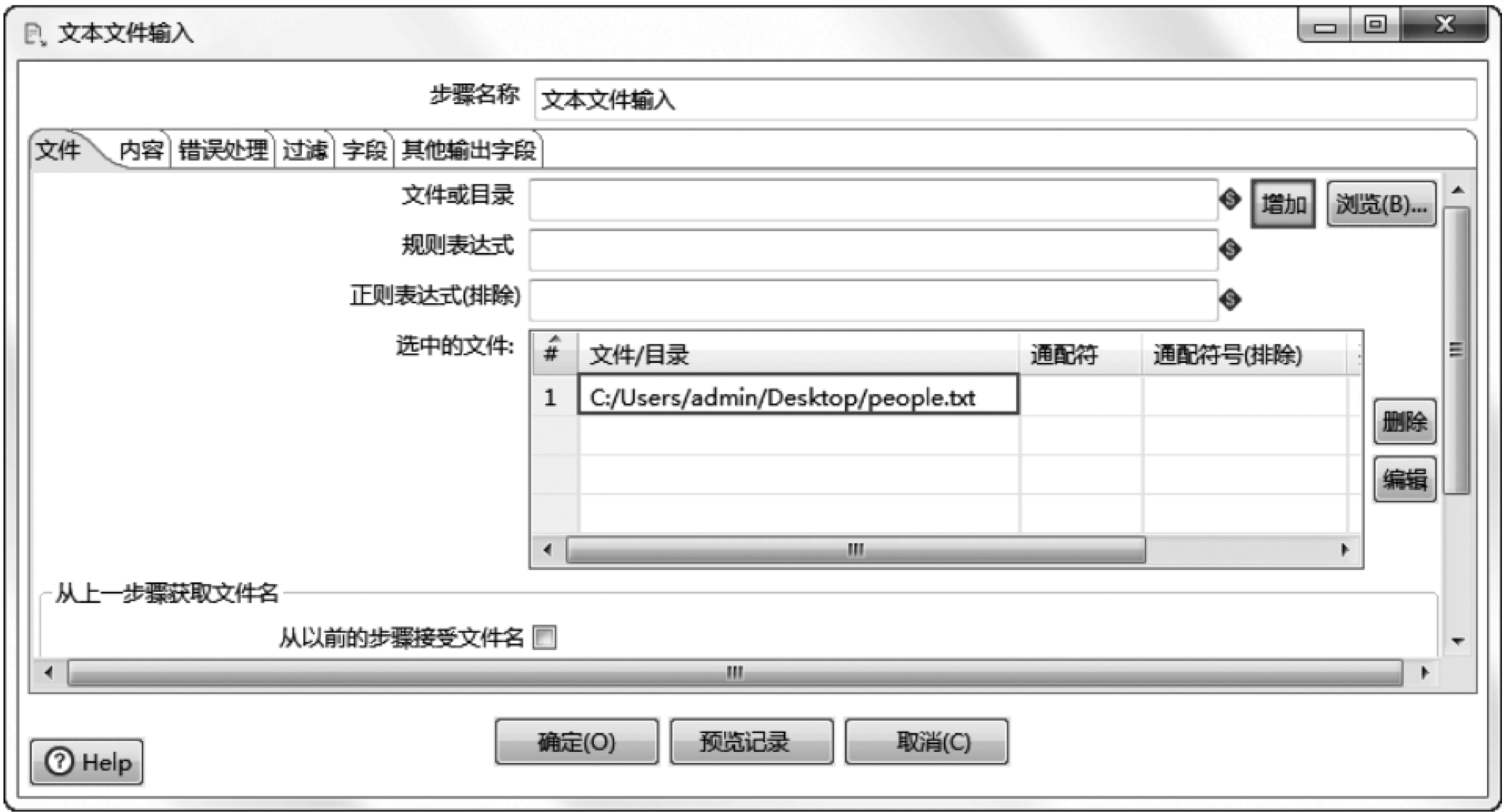


图 5-16　添加文件 people.txt 至转换 part_repeat_transform 中

在图 5-19 中，根据文件 people.txt 的内容添加对应的字段名称，并指定数据类型，这里需要注意的是，制表符可看作是由多个空格组成，因此在“去除空字符串方式”列时，所添加的字段都应选择“不去掉空格”，否则在抽取数据操作时，会把制表符当作空格去除，而不能把制表符作为分隔符实现文本文件内容的分隔。“字段”选项卡的配置如图 5-20 所示。

在图 5-20 中单击“预览记录”按钮，查看文件 people.txt 的数据是否成功抽取到文本文件输入流中，具体效果如图 5-21 所示。

从图 5-21 中可以看出，文件 people.txt 的数据已经成功抽取到文本文件输入流中，单击“关闭”→“确定”按钮，完成“文本文件输入”控件的配置。



图 5-17 “内容”选项卡界面



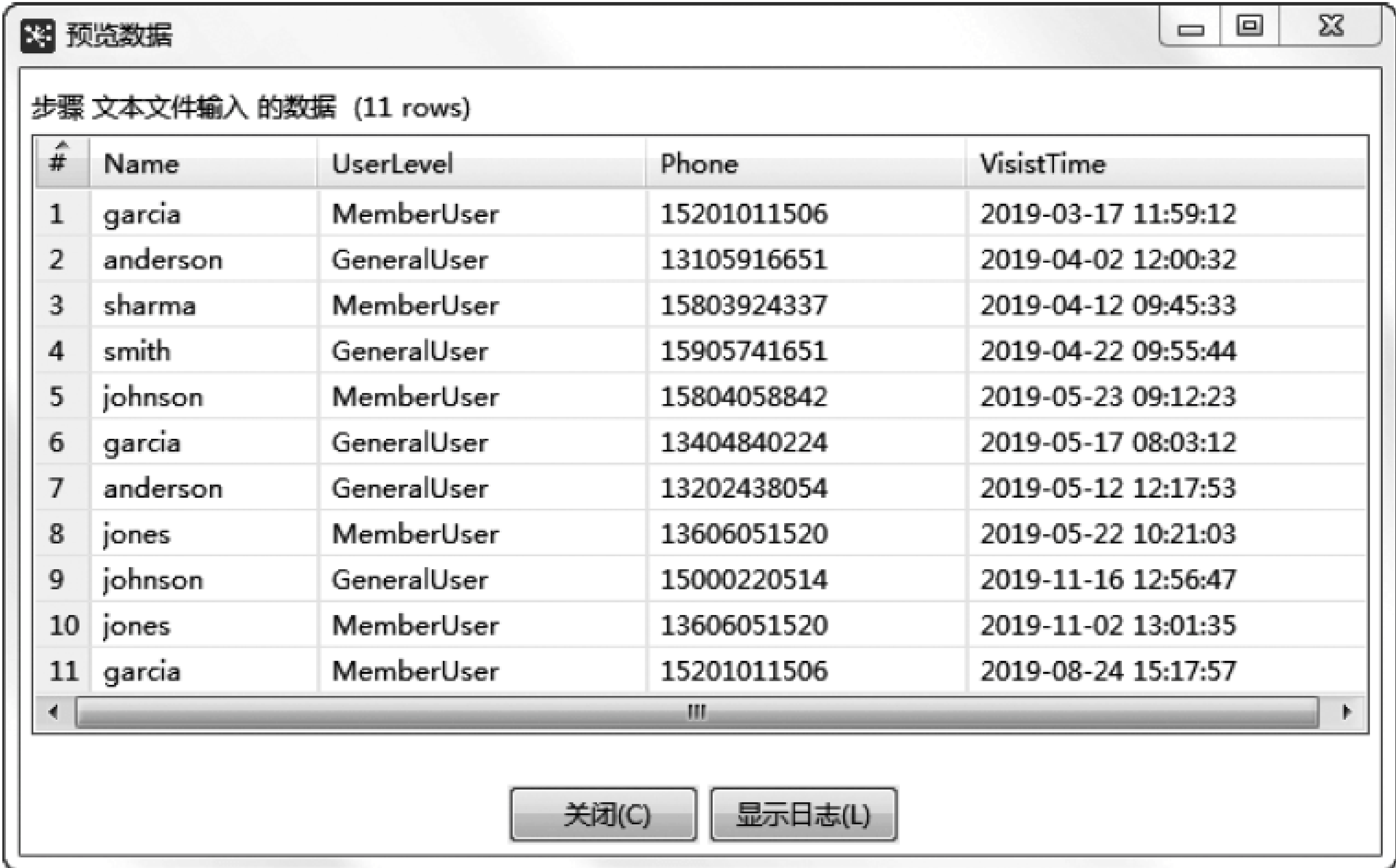
图 5-18 “内容”选项卡的配置



图 5-19 “字段”选项卡界面



图 5-20　“字段”选项卡的配置


预览数据

步骤 文本文件输入 的数据 (11 rows)

#	Name	UserLevel	Phone	VisistTime
1	garcia	MemberUser	15201011506	2019-03-17 11:59:12
2	anderson	GeneralUser	13105916651	2019-04-02 12:00:32
3	sharma	MemberUser	15803924337	2019-04-12 09:45:33
4	smith	GeneralUser	15905741651	2019-04-22 09:55:44
5	johnson	MemberUser	15804058842	2019-05-23 09:12:23
6	garcia	GeneralUser	13404840224	2019-05-17 08:03:12
7	anderson	GeneralUser	13202438054	2019-05-12 12:17:53
8	jones	MemberUser	13606051520	2019-05-22 10:21:03
9	johnson	GeneralUser	15000220514	2019-11-16 12:56:47
10	jones	MemberUser	13606051520	2019-11-02 13:01:35
11	garcia	MemberUser	15201011506	2019-08-24 15:17:57

关闭(C)　显示日志(L)

图 5-21　预览数据

3．配置“唯一行(哈希值)”控件

双击图 5-13 中的“唯一行(哈希值)”控件，进入“唯一行(哈希值)”界面，如图 5-22 所示。

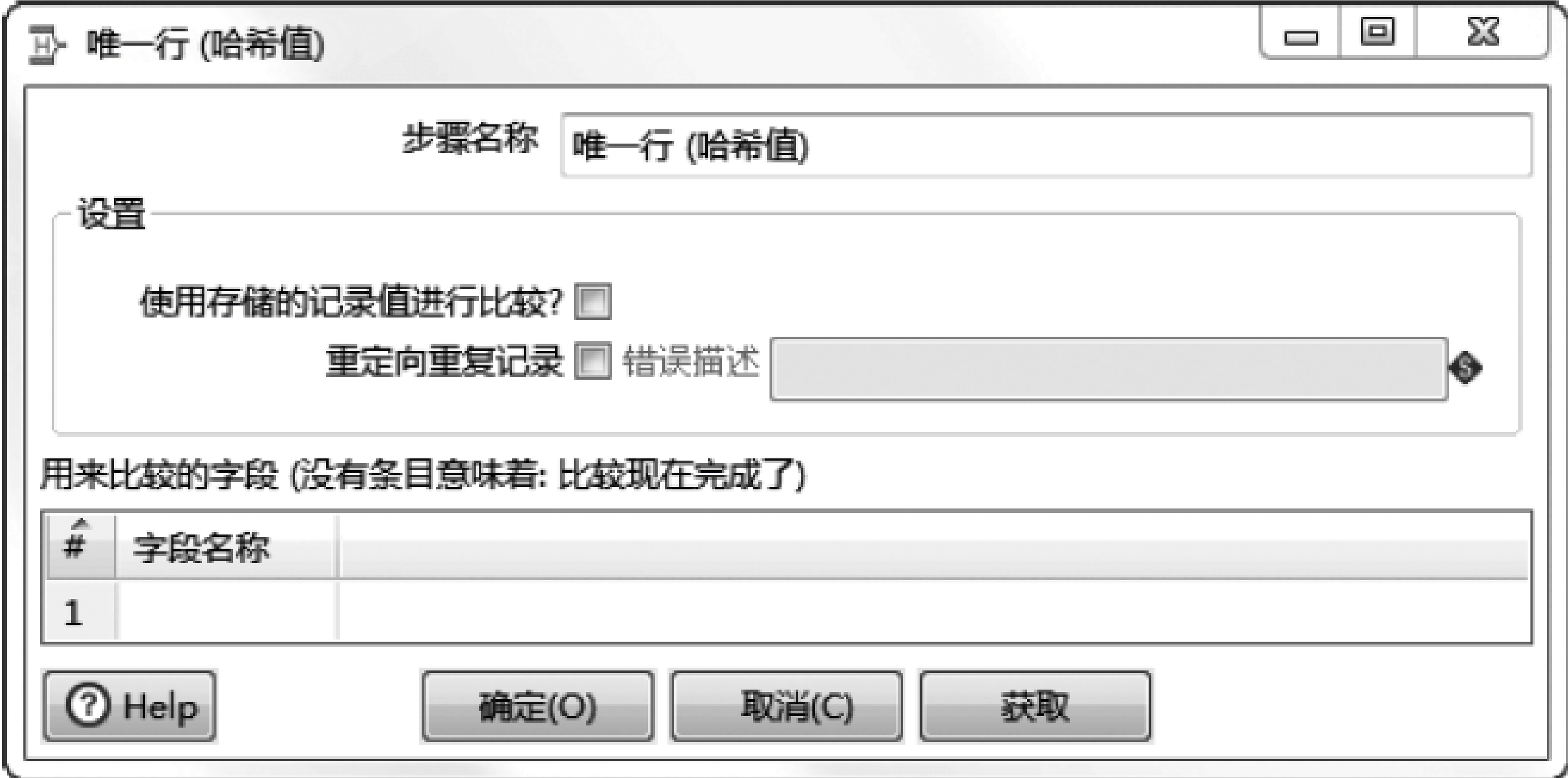


图 5-22　“唯一行(哈希值)”界面

在图 5-22 中的"用来比较的字段"处添加要比较去重的字段，即 Name、UserLevel、Phone 字段，具体如图 5-23 所示。

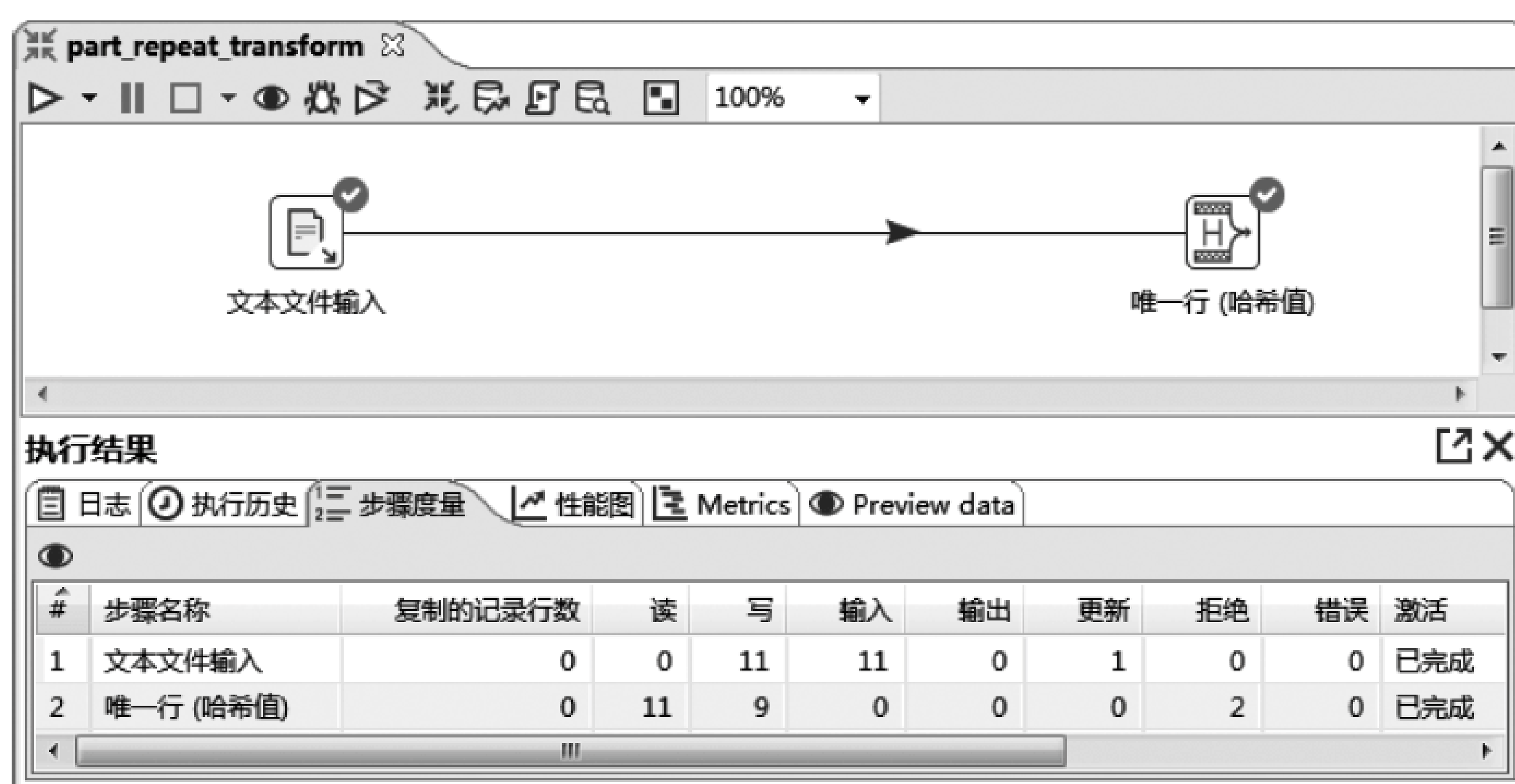


图 5-23　添加要比较去重的字段

在图 5-23 中单击"确定"按钮，完成"唯一行(哈希值)"控件的配置。

4. 运行转换 part_repeat_transform

单击转换工作区顶部的 ▷ 按钮，运行创建的转换 part_repeat_transform，实现文件 people.txt 中数据的不完全重复处理，具体如图 5-24 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	文本文件输入	0	0	11	11	0	1	0	0	已完成
2	唯一行 (哈希值)	0	11	9	0	0	0	2	0	已完成

图 5-24　运行转换 part_repeat_transform

从图 5-24 中执行结果窗口的"步骤度量"选项卡可以看出，"文本文件输入"控件输入 11 条数据并写入该控件中；"唯一行(哈希值)"控件读取"文本文件输入"控件中的 11 条数据，其中写入该控件 9 条数据，拒绝 2 条数据。也就是说，文件 people.txt 中有 2 条数据与其他数据是不完全重复的。

单击图 5-24 中的"唯一行(哈希值)"控件，再单击执行结果窗口的 Preview data 选项卡，查看是否消除文件 people.txt 中不完全重复的数据，具体如图 5-25 所示。

从图 5-25 中可以看出，文件 people.txt 中没有重复的数据，说明通过 Kettle 工具实现了消除不完全重复数据的功能。

执行结果

日志 | 执行历史 | 步骤度量 | 性能图 | Metrics | Preview data

${TransPreview.FirstRows.Label}　${TransPreview.LastRows.Label}　${TransPreview.Off.Label}

#	Name	UserLevel	Phone	VisistTime
1	garcia	MemberUser	15201011506	2019-03-17 11:59:12
2	anderson	GeneralUser	13105916651	2019-04-02 12:00:32
3	sharma	MemberUser	15803924337	2019-04-12 09:45:33
4	smith	GeneralUser	15905741651	2019-04-22 09:55:44
5	johnson	MemberUser	15804058842	2019-05-23 09:12:23
6	garcia	GeneralUser	13404840224	2019-05-17 08:03:12
7	anderson	GeneralUser	13202438054	2019-05-12 12:17:53
8	jones	MemberUser	13606051520	2019-05-22 10:21:03
9	johnson	GeneralUser	15000220514	2019-11-16 12:56:47

图 5-25　查看是否消除文件 people.txt 中不完全重复的数据

需要注意的是，在进行不完全重复数据去重时需要合理选择字段，否则有可能造成数据丢失的情况。上述的不完全去重例子中，如果只选择字段 Name 和字段 UserLevel，容易造成源文件 people.txt 中第 2 条数据和第 7 条数据丢失。

注意：本小节操作只是将文件 people.txt 的数据读取到 Kettle 中进行不完全去重处理，并不会改变文件 people.txt 的原始数据，如需保存处理后的数据，须添加相关输出控件。

5.2　缺失值处理

缺失值是指数据集中某个或某些属性的值是不完整的，产生的原因主要有人为原因和机械原因两种，其中机械原因是由于机器故障造成数据未能收集或存储失败，人为原因是由主观失误或有意隐瞒造成的数据缺失。本节将针对缺失值的处理进行详细讲解。

5.2.1　缺失值清洗策略

制定合理的缺失值数据处理策略，不仅可以提升缺失值数据处理的效率，还可以使处理后数据的可靠性得到保证，提高最终分析结果的准确性。缺失值的处理方法很多，这里建议大家在清洗缺失值时，首先计算数据源字段缺失值比例，之后根据数据缺失率和重要性制定不同的策略。下面通过一张图描述缺失数据的缺失率以及重要性划分的 4 种情况，具体如图 5-26 所示。

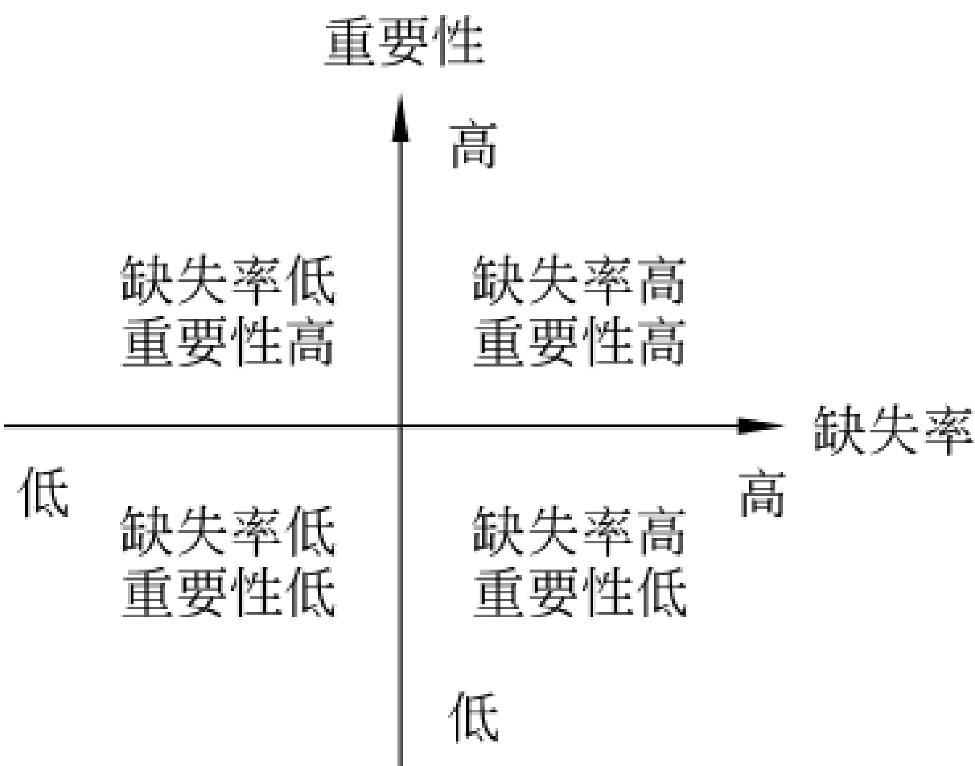


图 5-26　缺失值范围

根据图 5-26 确定缺失值的范围并制定对应的策略。根据缺失值的范围采用对应的策略见表 5-1。

表 5-1　根据缺失值的范围采用对应的策略

缺失值的范围	对应的策略
重要性高、缺失率高的数据	(1) 尝试从其他渠道获取数据进行补全； (2) 通过对其他字段的数据进行分析、计算等方式获取合理值进行补全； (3) 去除字段但要在结果中进行标注

续表

缺失值的范围	对应的策略
重要性高、缺失率低的数据	(1) 通过对字段自身的数据进行分析、计算等方式获取合理值进行补全； (2) 通过自身的经验与业务知识对缺失值数据进行人为补全
重要性低、缺失率高的数据	直接去除该字段
重要性低、缺失率低的数据	可以不处理或者进行简单的填充

5.2.2 去除缺失值

数据缺失分为两种：一种是行记录的缺失，这种情况又称数据记录丢失；另一种是列值的缺失，即由于各种原因导致的数据记录中某些列的值空缺。

去除缺失值数据通常有两种情况：一种是删除存在遗漏信息属性值的对象的列；另一种是删除存在遗漏信息属性值对象的记录，从而得到一个完备的信息表。缺失值数据在缺失值所在的列对最终分析结果无重要意义或存在缺失值的记录与初始数据集的数据量相比非常小的情况下非常有效。

在数据量小的情况下，通过人为观察就可以轻易从数据集中找到存在缺失值的记录；若数据量比较大，那么通过人为观察的方式查找存在缺失值的记录是非常耗时的，因此，可利用统计学方法筛选出包含缺失值的对象，然后通过计算得出每个字段的缺失率，去除缺失率高的字段，最后再对数据进行过滤，将有缺失值的记录过滤掉，这样就可以避免数据大量丢失。

假设现在有一份就业人员的收入数据文件 revenue.txt，由于某种原因，在数据采集的过程中产生了大量的缺失值数据。文件 revenue.txt 的具体内容如图 5-27 所示。

```
garcia   Female  5000    California
anderso  Female  6000    California
sharma           4000    New Jersey
smith    Female  8000    New York
johnson          9000    New York
garcia   Female  10000   New York
anderso          12000   California
jones    Male    3000    New York
johnson          5500    New York
jones            6000    California
garcia   Female          California
```

图 5-27 文件 revenue.txt 的具体内容

下面通过 Kettle 工具分步骤讲解如何去除原始数据集中的缺失值，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 delete_missing_value，并添加“文本文件输入”控件、“字段选择”控件、“过滤记录”控件、“Excel 输出”控件、“空操作(什么也不做)”控件以及 Hop 跳连接线，具体效果如图 5-28 所示。

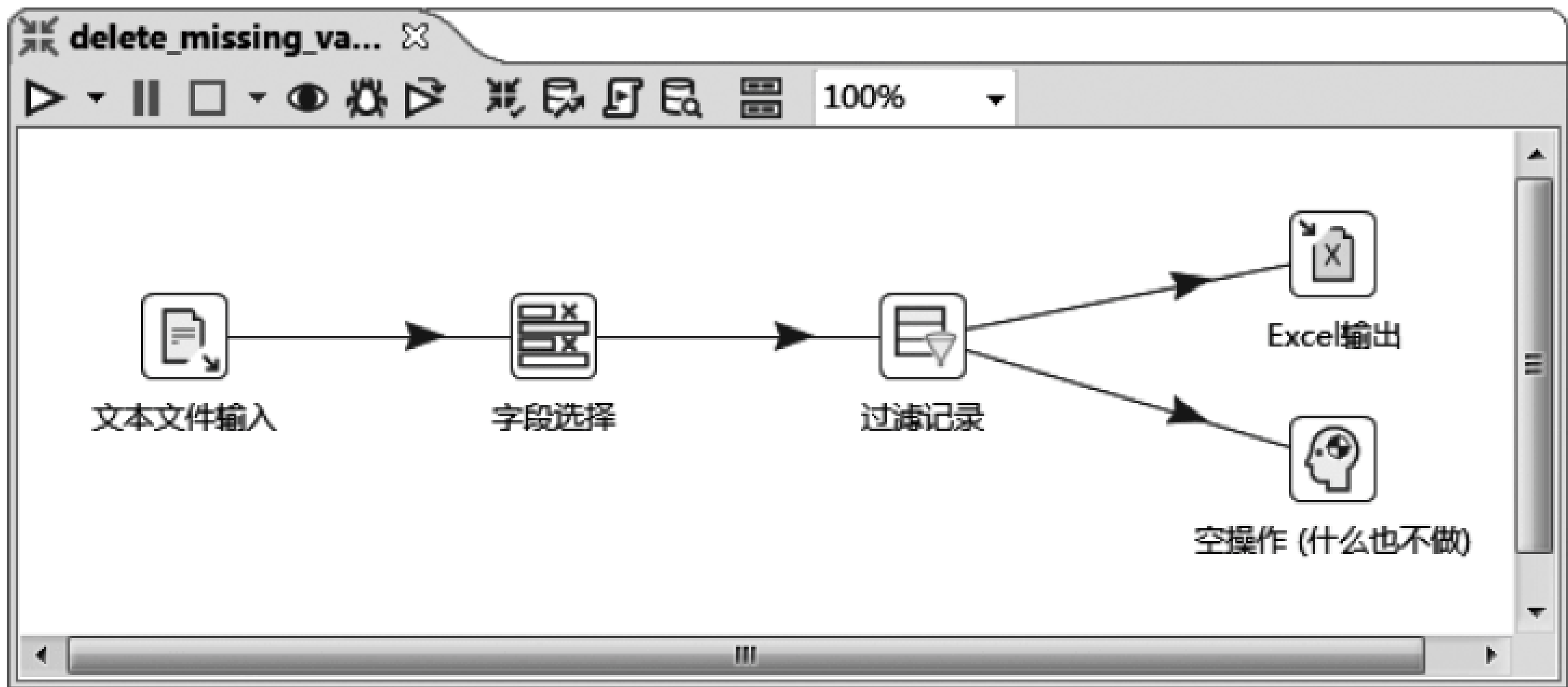


图 5-28 创建转换 delete_missing_value

2. 配置"文本文件输入"控件

双击图 5-28 中的"文本文件输入"控件,进入"文本文件输入"界面,如图 5-29 所示。



图 5-29 "文本文件输入"界面

在图 5-29 中单击"浏览"按钮,选择要去除缺失值的文件 revenue. txt,效果如图 5-30 所示。

在图 5-30 中单击"增加"按钮,将要去除缺失值的文件 revenue. txt 添加到"文本文件输入"控件中,具体效果如图 5-31 所示。

在图 5-31 中单击"内容"选项卡,切换到"内容"选项卡界面,如图 5-32 所示。

在图 5-32 中清除分隔符处的默认分隔符";",单击 Insert TAB 按钮,在分隔符处插入一个制表符;取消勾选"头部"复选框,若不取消,在进行数据抽取操作时会排除文件第一行的数据。"内容"选项卡的配置如图 5-33 所示。

在图 5-33 中单击"字段"选项卡,切换到"字段"选项卡界面,如图 5-34 所示。

在图 5-34 中,根据文件 revenue. txt 的内容添加对应的字段名称,并指定数据类型。需要注意的是,制表符可看作是由多个空格组成,因此在"去除空字符串方式"列时,所添加的



图 5-30 选择要去除缺失值的文件 revenue. txt



图 5-31 添加文件 revenue. txt 至控件“文本文件输入”



图 5-32 “内容”选项卡界面



图 5-33　“内容”选项卡的配置



图 5-34　“字段”选项卡界面

字段都应选择“不去掉空格”，否则在抽取数据操作时会把制表符当作空格去除，而不能把制表符作为分隔符实现文本文件内容的分隔，如图 5-35 所示。



图 5-35　添加字段

在图 5-35 中单击“预览记录”按钮，查看文件 revenue. txt 中的数据是否成功抽取到文

本文件输入流中，具体效果如图 5-36 所示。

预览数据

步骤 文本文件输入 的数据 (11 rows)

#	Name	Sex	Pay	Area
1	garcia	Female	5000	California
2	anderso	Female	6000	California
3	sharma	<null>	4000	New Jersey
4	smith	Female	8000	New York
5	johnson	<null>	9000	New York
6	garcia	Female	10000	New York
7	anderso	<null>	12000	California
8	jones	Male	3000	New York
9	johnson	<null>	5500	New York
10	jones	<null>	6000	California
11	garcia	Female	<null>	California

关闭(C)　显示日志(L)

图 5-36　预览数据

从图 5-36 中可以看出，文件 revenue.txt 中的数据已经成功抽取到文本文件输入流中，单击“关闭”→“确定”按钮，完成“文本文件输入”控件的配置。

3. 配置“字段选择”控件

双击图 5-28 中的“字段选择”控件，进入“选择/改名值”界面，如图 5-37 所示。

选择/改名值

步骤名称　字段选择

选择和修改　移除　元数据

字段：

#	字段名称	改名成	长度	精度
1				

获取选择的字段　列映射

包含未指定的列,按名称排序

Help　确定(O)　取消(C)

图 5-37　“选择/改名值”界面

在图 5-37 中“选择和修改”选项卡的“字段名称”处手动添加文本文件输入控件输出的所有数据字段，也可以单击“获取选择的字段”按钮，Kettle 工具自动检索并添加文本文件输入控件输出的所有数据字段，具体如图 5-38 所示。

在图 5-38 的“移除”选项卡中添加要移除的字段名称，这里移除的是 Sex 字段，如图 5-39 所示。

在图 5-39 中单击“确定”按钮，完成“字段选择”控件的配置。

4. 配置“过滤记录”控件

双击图 5-28 中的“过滤记录”控件，进入“过滤记录”界面，如图 5-40 所示。



图 5-38　添加字段



图 5-39　添加要移除的字段名称



图 5-40　“过滤记录”界面

在图 5-40 中的“条件”处设置过滤的条件，过滤掉有缺失值的数据字段(这里是过滤 Name、Pay 和 Area 字段中的缺失值)。单击左边的<field>框，弹出“字段”对话框，选择要过滤的字段 Name，具体如图 5-41 所示。

在图 5-41 中单击“确定”按钮，完成过滤字段 Name 的选择。

单击图 5-40 中的“=”框，弹出“函数:”对话框，选择过滤条件(这里选择的是 IS NULL)，具体如图 5-42 所示。



图 5-41　“字段”对话框

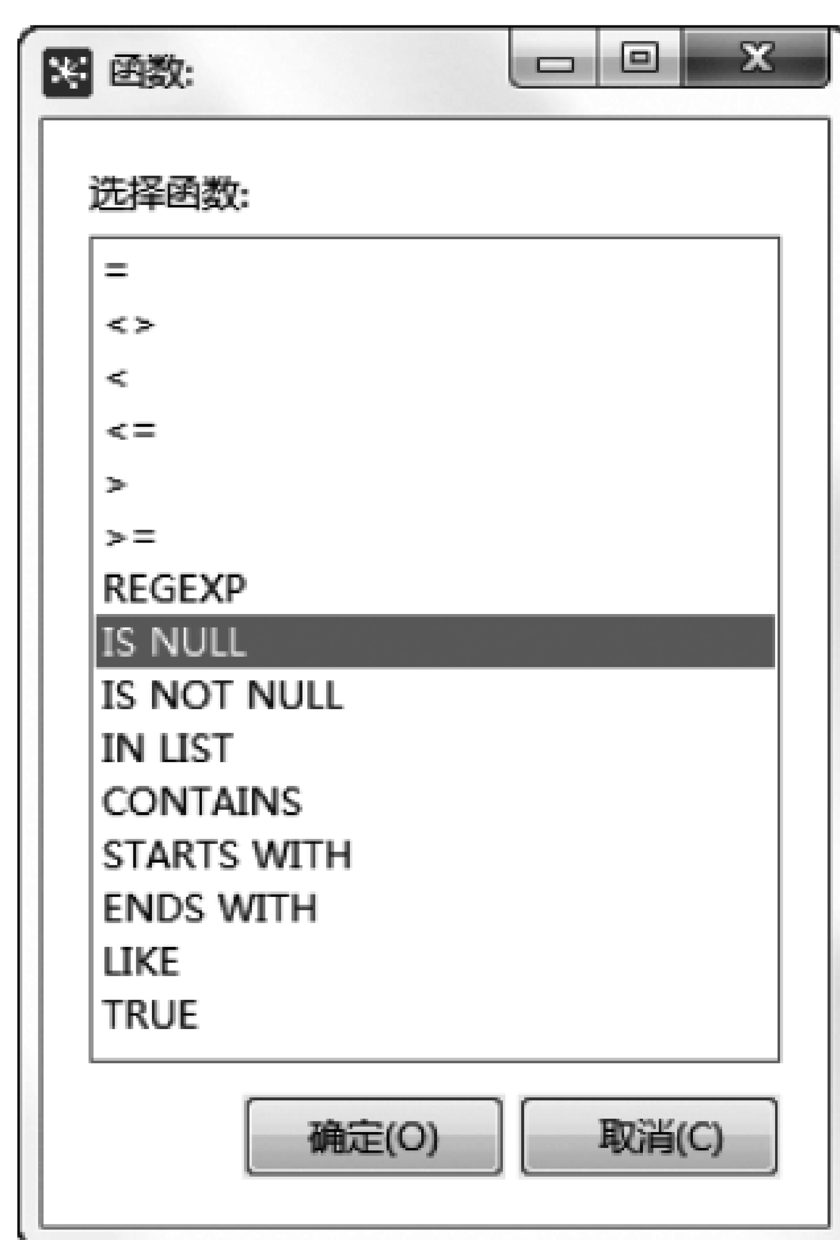


图 5-42　“函数:”对话框

在图 5-42 中单击“确定”按钮，完成过滤条件的选择。字段 Name 的过滤设置如图 5-43 所示。



图 5-43　字段 Name 的过滤设置

在图 5-43 中单击符号“＋”增加过滤条件，具体如图 5-44 所示。



图 5-44　增加过滤条件

在图 5-44 中单击 AND，弹出“操作符:”对话框，选择操作符（这里选择的是 OR），如图 5-45 所示。

在图 5-45 中单击“确定”按钮，完成操作符的选择。

单击图 5-44 中的“null = [ ]”，添加过滤字段，具体如图 5-46 所示。



图 5-45　选择操作符　　　　**图 5-46　添加过滤字段**

在图 5-46 中单击左边的＜field＞框，弹出“字段”对话框，选择要过滤的字段 Pay；单击“＝”框，弹出“函数:”对话框，选择过滤条件（这里选择的是 IS NULL）。字段 Pay 的过滤设置如图 5-47 所示。



图 5-47　字段 Pay 的过滤设置

在图 5-47 中单击符号“＋”增加过滤条件，具体如图 5-48 所示。

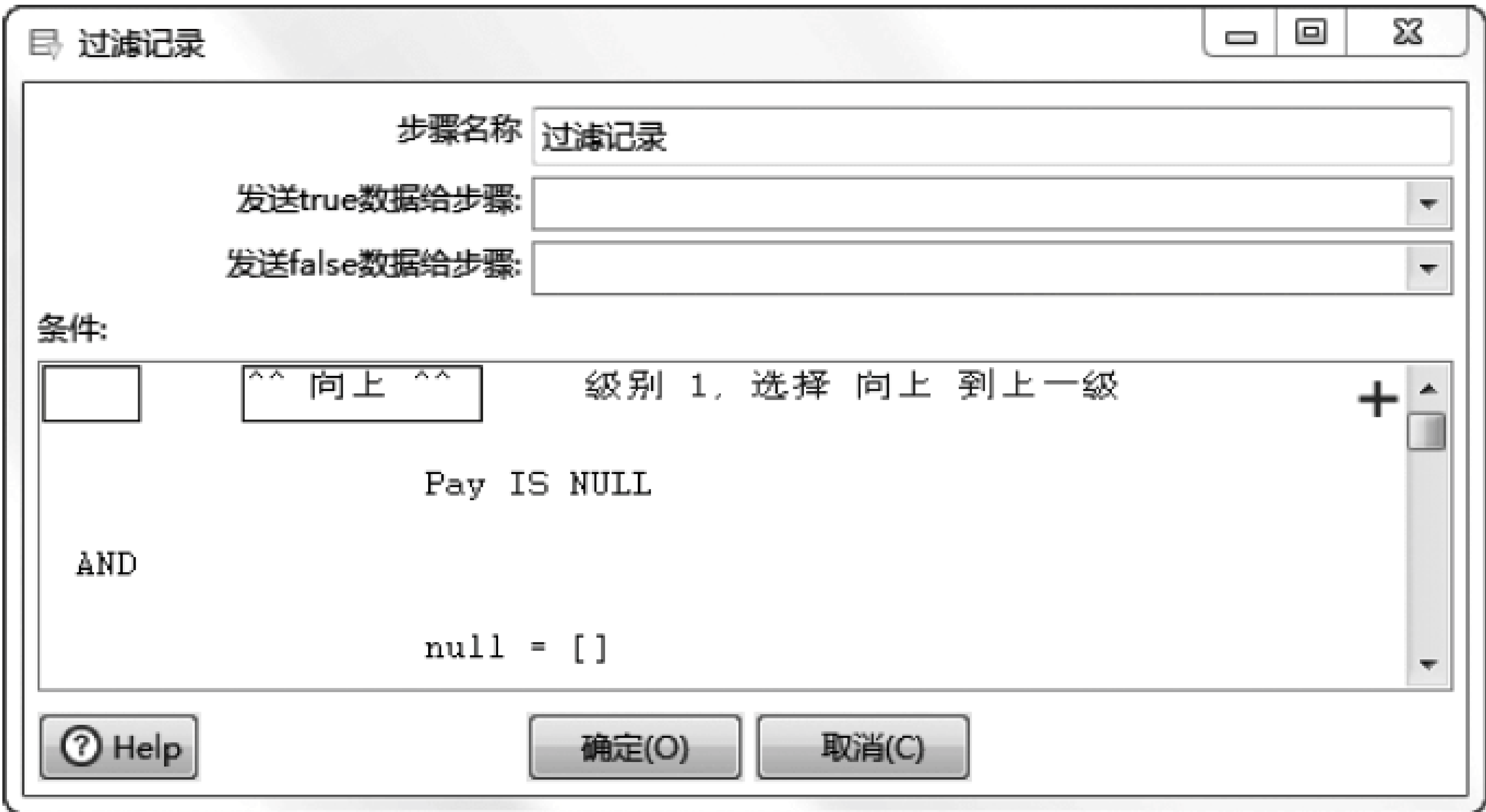


图 5-48　增加过滤条件

在图 5-48 中单击 AND，弹出“操作符:”对话框，选择操作符（这里选择的是 OR），如图 5-49 所示。

在图 5-49 中单击“确定”按钮，完成操作符的选择。

单击图 5-48 中的“null = [ ]”，添加过滤字段，具体如图 5-50 所示。



图 5-49　选择操作符



图 5-50　添加过滤字段

在图 5-50 中单击左边的＜field＞框，弹出“字段”对话框，选择要过滤的字段 Area；单击“=”框，弹出“函数:”对话框，选择过滤条件（这里选择的是 IS NULL）。字段 Area 的过滤设置如图 5-51 所示。



图 5-51　字段 Area 的过滤设置

在图 5-51 中连续单击两次“确定”按钮，查看设置的过滤条件，如图 5-52 所示。

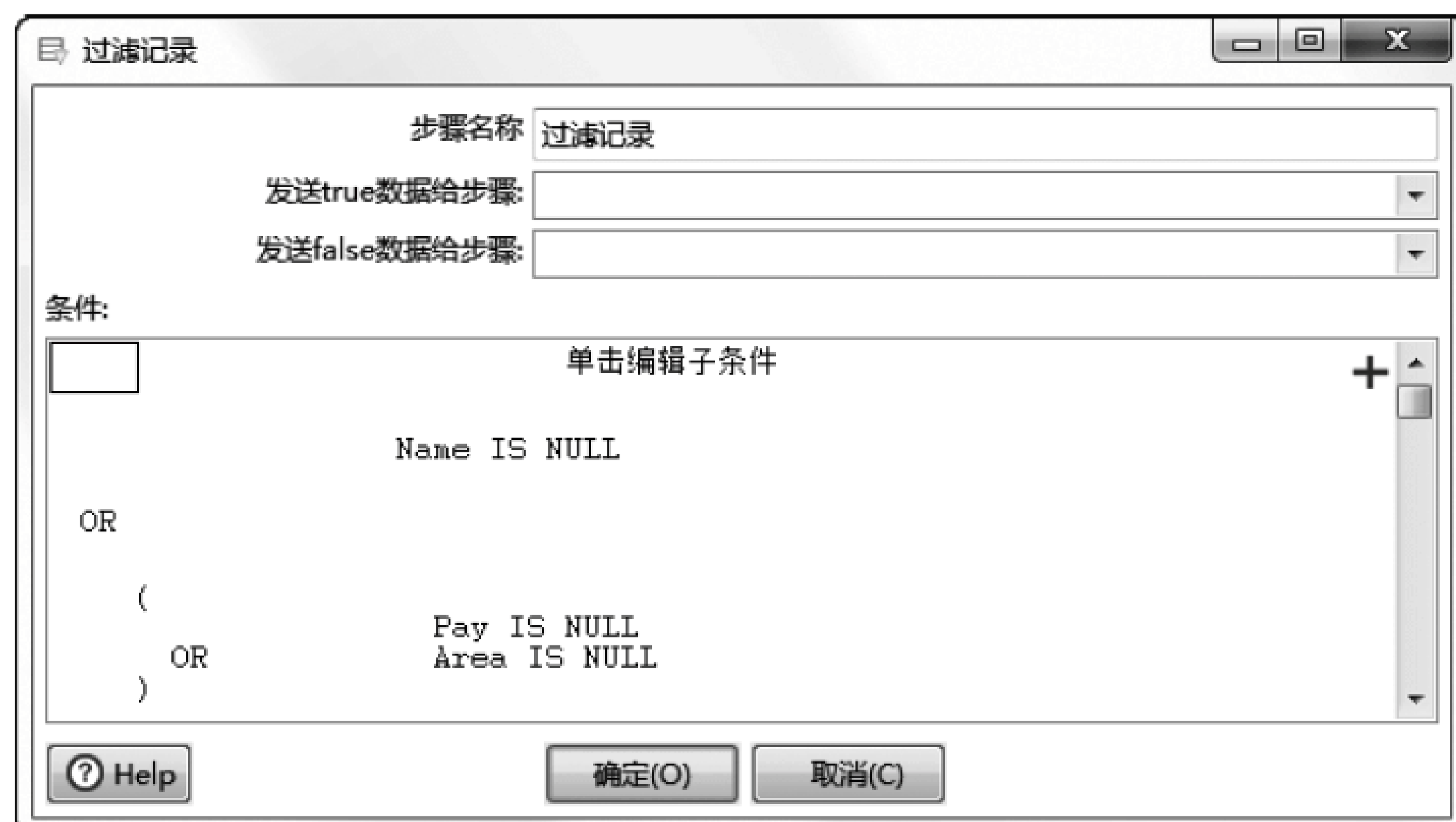


图 5-52　设置的过滤条件

在图 5-52 中“发送 true 数据给步骤：”后的下拉列表中选择“空操作”，将包含缺失值的行数据放在“空操作”控件中；在“发送 false 数据给步骤：”后的下拉列表中选择“Excel 输出”，将没有缺失值的行数据输出到 Excel 文件中，具体如图 5-53 所示。

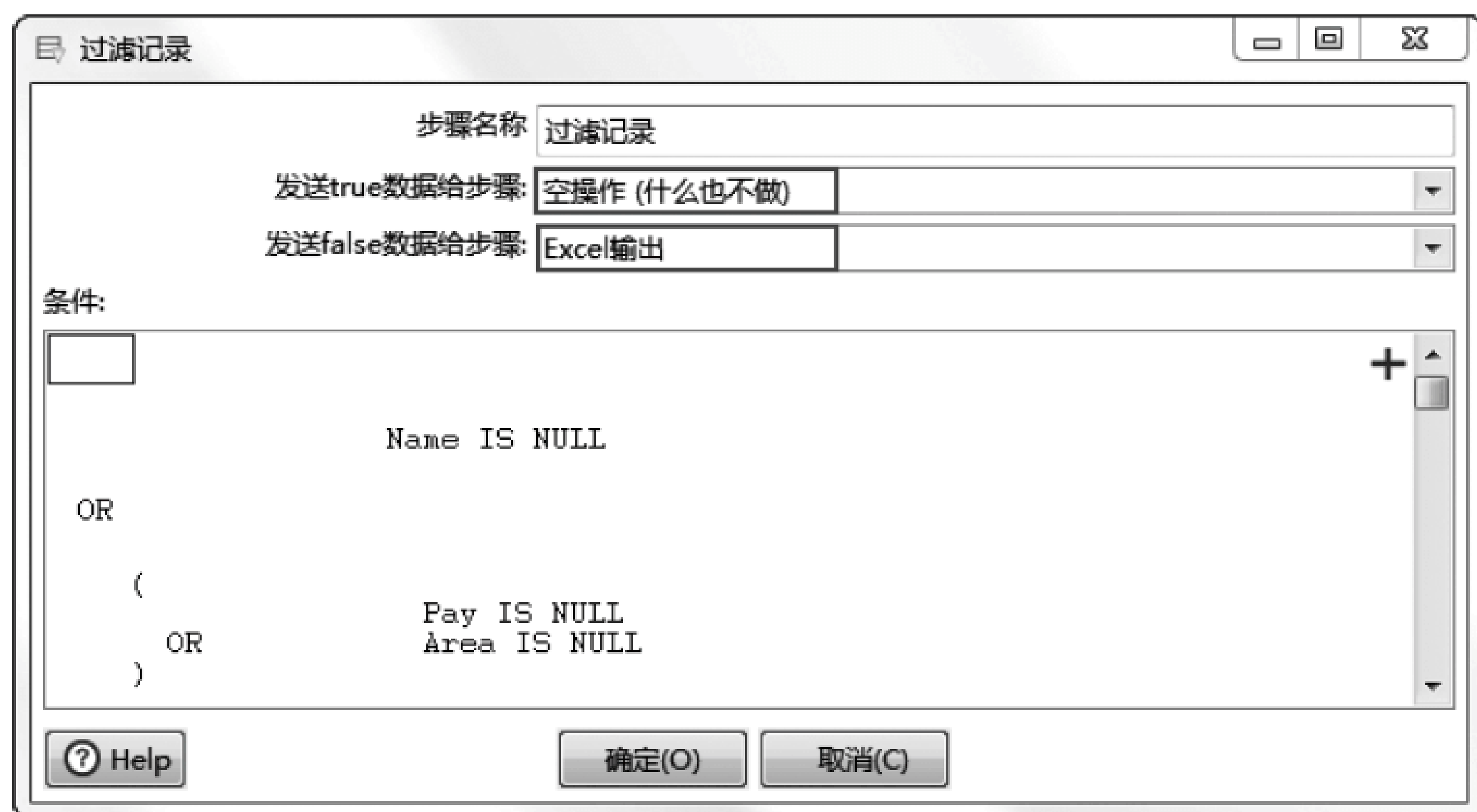


图 5-53　发送 true/false 数据给相关步骤的配置

在图 5-53 中单击“确定”按钮，完成“过滤记录”控件的配置。

5. 配置“Excel 输出”控件

双击图 5-28 中的“Excel 输出”控件，进入“Excel 输出”界面，如图 5-54 所示。



图 5-54　“Excel 输出”界面

在图 5-54 中单击“浏览”按钮，选择要输出的文件路径，如图 5-55 所示。

在图 5-55 中单击“确定”按钮，完成“Excel 输出”控件的配置。

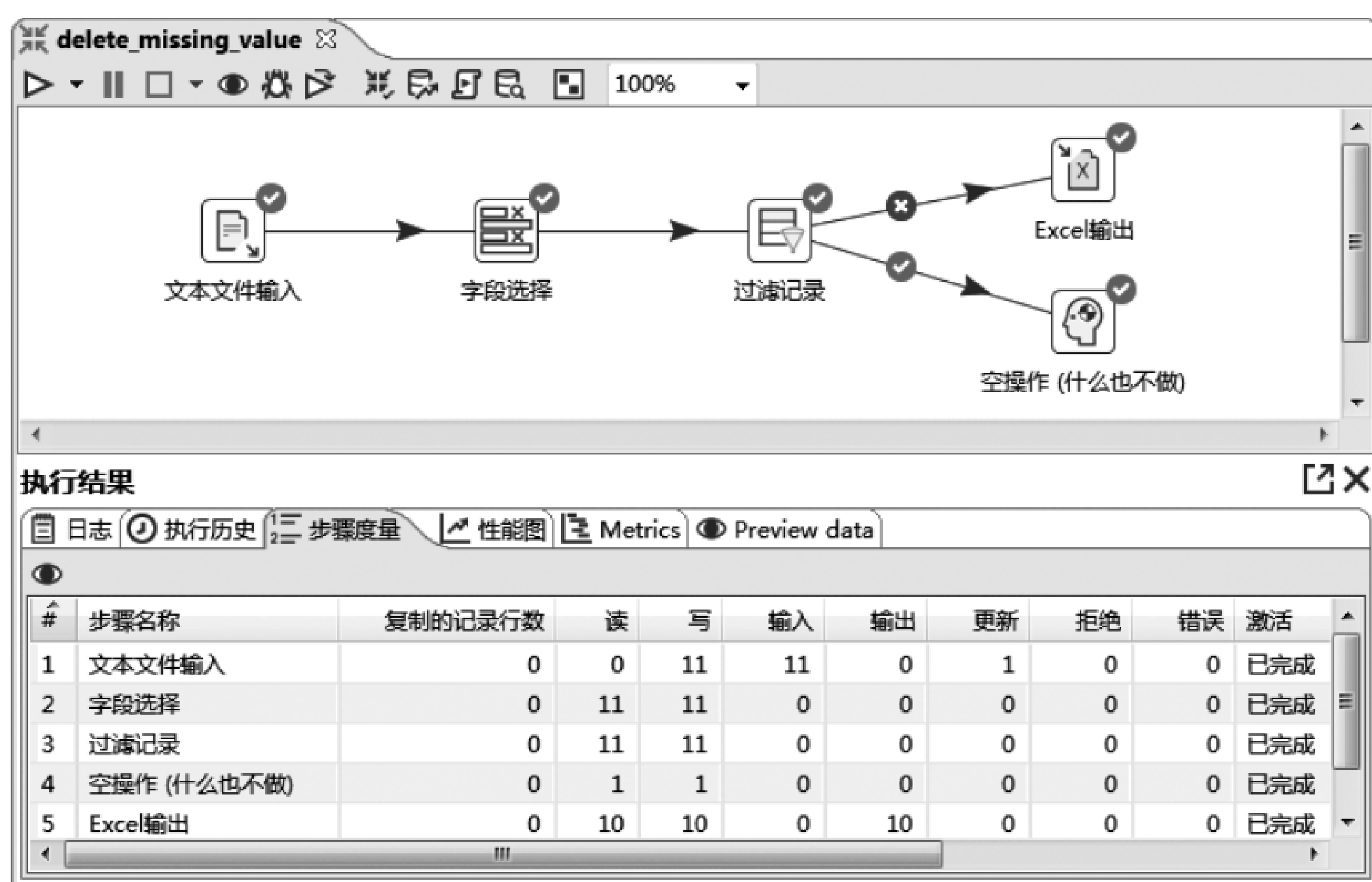
6. 运行转换 delete_missing_value

单击转换工作区顶部的▷按钮，运行创建的转换 delete_missing_value，实现去除文件 revenue. txt 中的缺失值，具体如图 5-56 所示。

从图 5-56 中执行结果窗口的“步骤度量”选项卡可以看出，“文本文件输入”控件输入 11 条数据并写入该控件；“字段选择”控件读取“文本文件输入”控件的 11 条数据并写入该控件；“过滤记录”控件读取“字段选择”控件的 11 条数据并写入该控件；通过条件过滤操作使



图 5-55　选择要输出的文件路径



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	文本文件输入	0	0	11	11	0	1	0	0	已完成
2	字段选择	0	11	11	0	0	0	0	0	已完成
3	过滤记录	0	11	11	0	0	0	0	0	已完成
4	空操作 (什么也不做)	0	1	1	0	0	0	0	0	已完成
5	Excel输出	0	10	10	0	10	0	0	0	已完成

图 5-56　运行转换 delete_missing_value

“空操作”控件读取并写入 1 条数据，“Excel 输出”控件读取并写入 10 条数据，最终输出。

7. 查看文件 file.xls

查看“Excel 输出”控件输出的文件 file.xls 是否还含有缺失值数据。文件 file.xls 的内容如图 5-57 所示。

从图 5-57 中可以看出，文件中的数据没有缺失值，说明我们已经完全去除了原始数据集(即文件 revenue.txt)中的缺失值。

5.2.3 填充缺失值

数据挖掘中面对的通常都是大型的数据库，它的属性有几十个甚至几百个，因为其中某个属性值的缺失而放弃大量其他的属性值，这种删除是对信息的极大浪费，所以产生了插补缺失值的思想与方法。常用的填充缺失值方法具体如下。

	A	B	C
1	Name	Pay	Area
2	garcia	5,000.00	California
3	anderso	6,000.00	California
4	sharma	4,000.00	New Jersey
5	smith	8,000.00	New York
6	johnson	9,000.00	New York
7	garcia	10,000.00	New York
8	anderso	12,000.00	California
9	jones	3,000.00	New York
10	johnson	5,500.00	New York
11	jones	6,000.00	California

图 5-57　文件 file. xls 的内容

1. 均值填充

数据的属性分为定距型和非定距型。如果缺失值是定距型的，就以该属性存在值的平均值插补缺失的值；如果缺失值是非定距型的，就根据统计学中的众数原理，用该属性的众数(即出现频率最高的值)补齐缺失的值。

2. 热卡填充

对于一个包含缺失值的对象，热卡填充方法会在完整数据集中找到一个与它最相似的对象的值进行填充。对于不同的问题，可能会选用不同的标准对相似对象进行判定，从概念上理解该方法很简单，利用数据间的关系进行缺失值评估。热卡填充方法的缺点在于难以定义相似标准，人为主观因素较多。

3. 回归填充

将缺失值变量(自身字段)作为因变量，相关变量(其他字段)作为自变量进行回归拟合，用预测值作为填补值，需要注意的是自变量的数据尽量是完整的。

与前述几种插补方法比较，回归填充方法的优势是可充分利用数据库中的信息，弊端主要有两点：第一，该方法是无偏估计，但容易忽视随机误差、低估标准差和其他未知性质的测量值，而且这一问题会随着缺失信息的增多变得更加严重；第二，研究者必须假设存在缺失值所在的变量与其他变量存在线性关系，大多数情况下这种关系是不存在的。

4. 多重填充

多重估算是由 Rubin 等人于 1987 年建立起的一种数据扩充和统计分析方法，作为简单估算的改进产物。首先，多重估算技术用一系列可能的值替换每个缺失值，以反映被替换的缺失数据的不确定性。然后，用标准的统计分析过程对多次替换后产生的若干个数据集进行分析。最后，把来自各个数据集的统计结果进行综合，得到总体参数的估计值。

由于多重估算技术并不是用单一的值替换缺失值，而是试图产生缺失值的一个随机样本，这种方法可以反映出由于数据缺失而导致的不确定性，产生更加有效的统计推断。

假设现在有一份社会人员调查信息的数据文件 people_survey. txt，由于某种原因，在

数据采集的过程中产生了大量的缺失值，文件 people_survey. txt 的内容如图 5-58 所示。

```
000001  40
000002  18  Private
000003  40
000004  40  Private
000005  40  Private
000006  45  Private
000007  40  Private
000008  20  State-gov
000009  40  Federal-gov
000010  60  Private
000011  35  Private
000012  45  Self-emp-not-inc
000013  20  Private
000014  55  Private
000015  40
000016      Private
000017  76  Private
000018  50  Private
000019  40  Private
000020  50  Private
000021  40  Private
```

图 5-58　文件 people_survey. txt 的内容

下面通过 Kettle 工具，分步骤讲解使用平均值填充法对文件 people_survey. txt 中的缺失值进行填充，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 fill_missing_value，并添加"文本文件输入"控件、"过滤记录"控件、"空操作（什么也不做）"控件、"替换 NULL 值"控件、"合并记录"控件、"字段选择"控件以及 Hop 跳连接线，具体效果如图 5-59 所示。

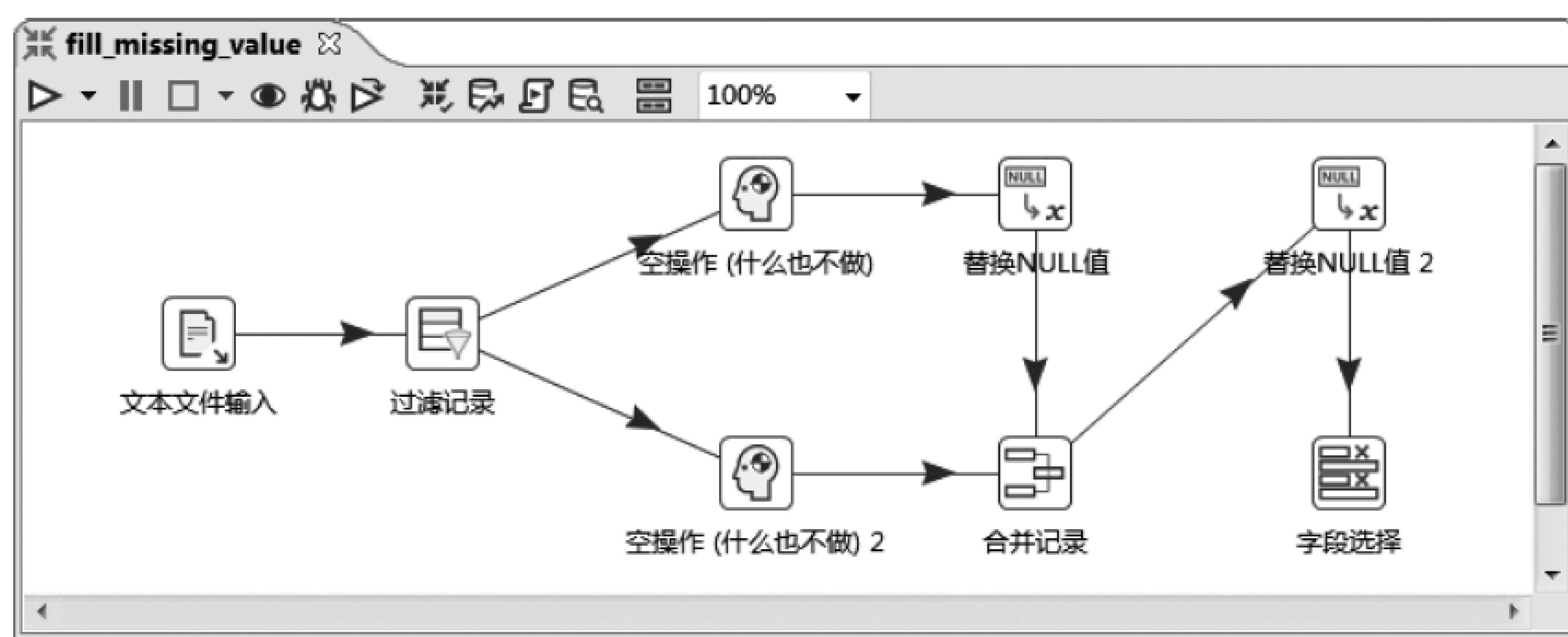


图 5-59　转换 fill_missing_value 的内容

2. 配置"文本文件输入"控件

双击图 5-59 中的"文本文件输入"控件，进入"文本文件输入"界面，如图 5-60 所示。

在图 5-60 中单击"浏览"按钮，选择要填充缺失值的文件 people_survey. txt，效果如图 5-61 所示。



图 5-60　“文本文件输入”界面

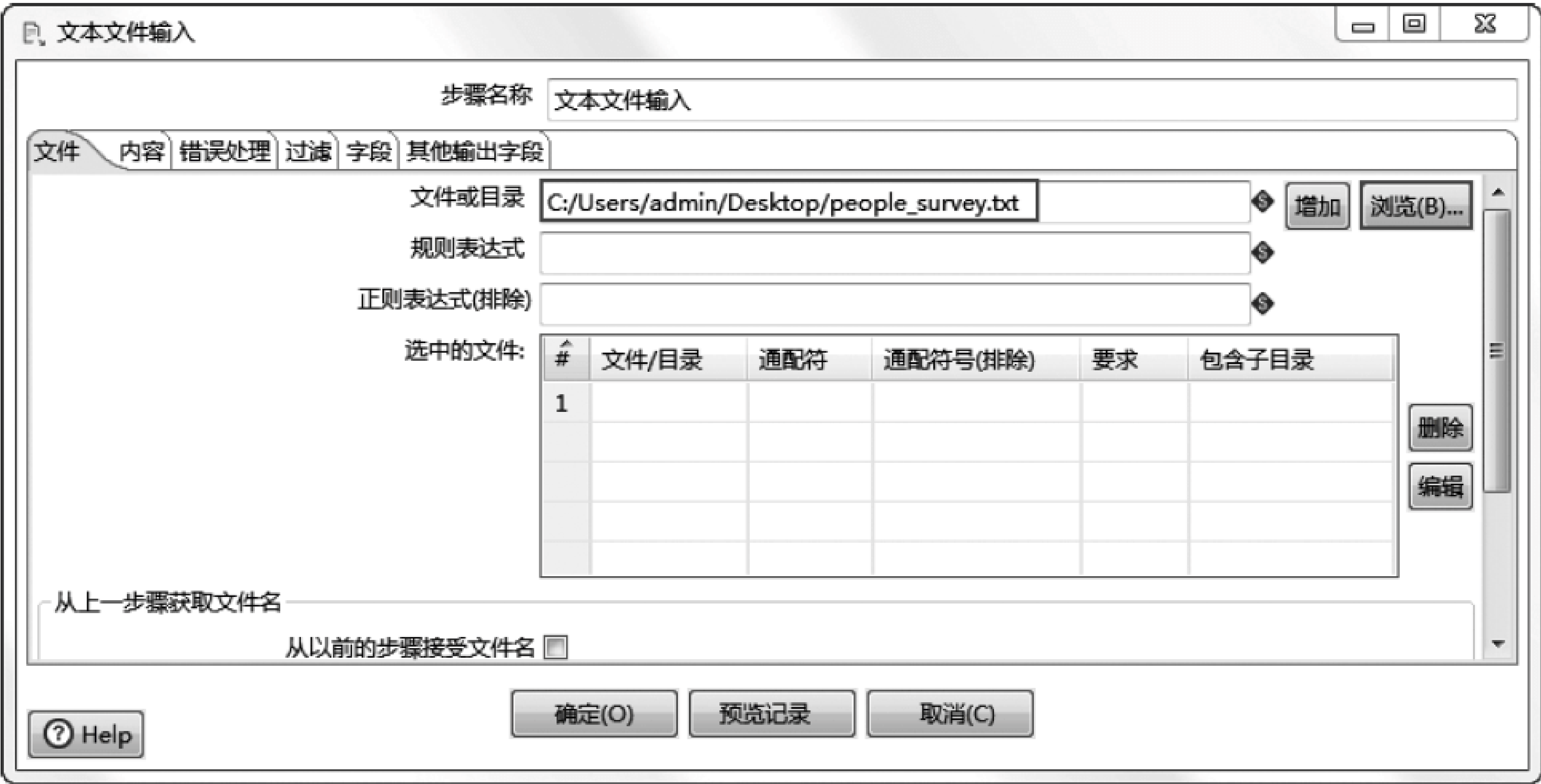


图 5-61　选择要填充缺失值的文件 people_survey. txt

在图 5-61 中单击“增加”按钮，将要填充缺失值的文件 people_survey. txt 添加到“文本文件输入”控件中，具体效果如图 5-62 所示。

在图 5-62 中单击“内容”选项卡，切换到“内容”选项卡界面，如图 5-63 所示。

在图 5-63 中清除分隔符处的默认分隔符“;”，单击 Insert TAB 按钮，在分隔符处插入一个制表符；取消勾选“头部”复选框，若不取消，则在进行数据抽取操作时会排除文件第一行的数据。“内容”选项卡的配置如图 5-64 所示。

在图 5-64 中单击“字段”选项卡，切换到“字段”选项卡界面，如图 5-65 所示。

在图 5-65 中根据文件 people_survey. txt 的内容添加对应的字段名称，并指定数据类型。需要注意的是，制表符可看作是由多个空格组成，因此在“去除空字符串方式”列时，所添加的字段都应选择“不去掉空格”，否则在抽取数据操作时会把制表符当作空格去除，而不



图 5-62　添加文件 people_survey.txt 至“文本文件输入”控件中



图 5-63　“内容”选项卡界面



图 5-64　“内容”选项卡的配置



图 5-65　“字段”选项卡界面

能把制表符作为分隔符实现文本文件内容的分隔，如图 5-66 所示。



图 5-66　添加字段

在图 5-66 中单击“预览记录”按钮，查看文件 people_survey. txt 的数据是否成功抽取到文本文件输入流中，具体效果如图 5-67 所示。

从图 5-67 中可以看出，文件 people_survey. txt 的数据已经成功抽取到文本文件输入流中，单击“关闭”→“确定”按钮，完成“文本文件输入”控件的配置。

将字段 workclass 同为 Private 值的字段 hours_per_week 值相加求均值，并用该均值对字段 userid 中值为 000016 的 hours_per_week 字段存在的缺失值进行填充。

3. 配置“过滤记录”控件

双击图 5-59 中的“过滤记录”控件，进入“过滤记录”界面，如图 5-68 所示。

在图 5-68 中的“条件”处设置过滤的条件，由于从图 5-67 中可以看出字段 userid 为 000016 用户的 hours_per_week(即每周工作时间字段)存在缺失值，而它的 workclass 字段值为 Private，因此可以将过滤字段设置为 workclass、过滤值设置为 Private 作为过滤条件，具体如图 5-69 所示。

在图 5-69 中“发送 true 数据给步骤：”下拉列表中选择“空操作(什么也不做)”，将 workclass 字段值为 Private 的数据放在“空操作”控件中；在“发送 false 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)2”，将 workclass 字段值不为 Private 的数据放在“空

预览数据

步骤 文本文件输入 的数据 (21 rows)

#	userid	hours_per_week	workclass
1	000001	40	<null>
2	000002	18	Private
3	000003	40	<null>
4	000004	40	Private
5	000005	40	Private
6	000006	45	Private
7	000007	40	Private
8	000008	20	State-gov
9	000009	40	Federal-gov
10	000010	60	Private
11	000011	35	Private
12	000012	45	Self-emp-not-inc
13	000013	20	Private
14	000014	55	Private
15	000015	40	<null>
16	000016	<null>	Private
17	000017	76	Private
18	000018	50	Private
19	000019	40	Private
20	000020	50	Private
21	000021	40	Private

关闭(C) 显示日志(L)

图 5-67 预览数据

过滤记录

步骤名称 过滤记录

发送true数据给步骤:

发送false数据给步骤:

条件:

<value>

Help 确定(O) 取消(C)

图 5-68 “过滤记录”界面

过滤记录

步骤名称 过滤记录

发送true数据给步骤:

发送false数据给步骤:

条件:

workclass =

Private (String)

Help 确定(O) 取消(C)

图 5-69 设置过滤条件

操作(什么也不做)2”控件中,具体如图 5-70 所示。



图 5-70　配置发送 true/false 数据给相关步骤

在图 5-70 中单击“确定”按钮,完成“过滤记录”控件的配置。

4. 配置“替换 NULL 值”控件

双击图 5-59 中的“替换 NULL 值”控件,进入“替换 NULL 值”界面,如图 5-71 所示。



图 5-71　“替换 NULL 值”界面

在图 5-71 中勾选“选择字段”复选框,并在“字段”框添加字段 hours_per_week,值替换为 44(44 是字段为 hours_per_week 中所有值相加求的均值,这里指用 44 替换字段 hours_per_week 中的 NULL 值),具体如图 5-72 所示。

在图 5-72 中单击“确定”按钮,完成“替换 NULL 值”控件的配置。



图 5-72 配置“替换 NULL 值”控件

5. 配置“合并记录”控件

双击图 5-59 中的“合并记录”控件，进入“合并行(比较)”界面，如图 5-73 所示。

在图 5-73 中“旧数据源：”后的下拉列表中选择“替换 NULL 值”，在“新数据源：”后的下拉列表中选择“空操作(什么也不做)2”；在“匹配的关键字：”部分添加关键字段，即 userid，具体如图 5-74 所示。



图 5-73 “合并行(比较)”界面



图 5-74 配置“合并记录”控件

“合并记录”控件主要是将两个数据源(旧数据源、新数据源)进行合并，标志字段主要是将每条数据进行标记，新数据源的数据会标记为 new，旧数据源的数据会标记为 deleted，若新、旧数据源中存在相同的关键字段设置的数据，则两个数据源进行合并后，只会保存从新数据源中获取的数据，并以 identical 进行标记。

在图 5-74 中单击“确定”按钮，完成“合并记录”控件的配置。

6. 配置“替换 NULL 值 2”控件

双击图 5-59 中的“替换 NULL 值 2”控件，进入“替换 NULL 值”界面，如图 5-75 所示。

图 5-75　“替换 NULL 值”界面

在图 5-75 中勾选“选择字段”复选框，并在“字段”框添加字段为 workclass，值替换为 Private(这里用 Private 替换字段 workclass 中的 NULL 值)，具体如图 5-76 所示。

图 5-76　配置“替换 NULL 值 2”控件

在图 5-76 中单击“确定”按钮，完成“替换 NULL 值 2”控件的配置。

7. 配置“字段选择”控件

双击图 5-59 中的“字段选择”控件，进入“选择/改名值”界面，如图 5-77 所示。



图 5-77 “选择/改名值”界面

在图 5-77 的“移除”选项卡界面中添加要移除的字段名称，这里移除的是字段 flagfield，如图 5-78 所示。



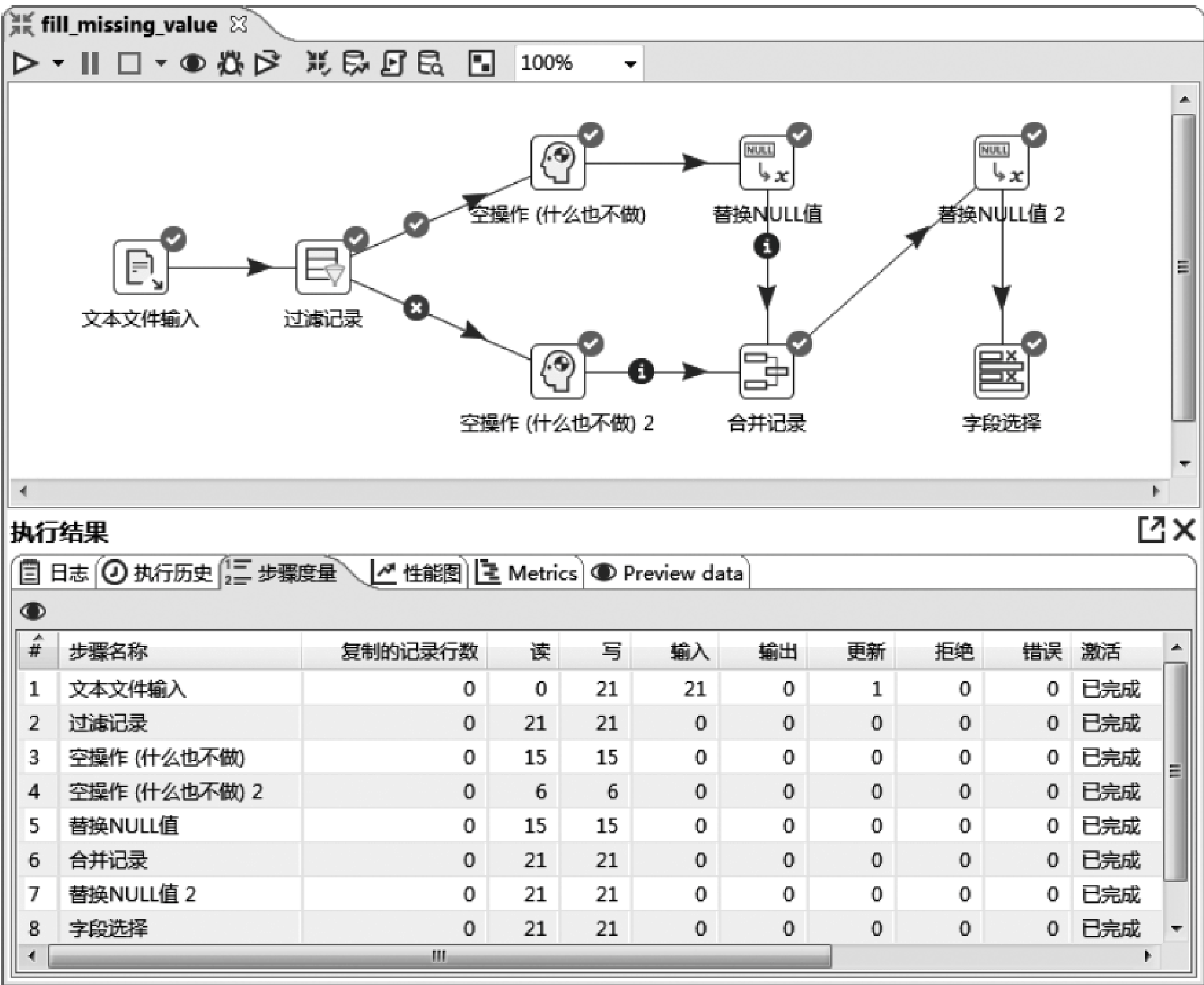
图 5-78 添加要移除的字段

在图 5-78 中单击“确定”按钮，完成“字段选择”控件的配置。

8. 运行转换 fill_missing_value

单击转换工作区顶部的 ▷ 按钮，运行创建的转换 fill_missing_value，实现填充文件 people_survey.txt 中的缺失值，具体如图 5-79 所示。

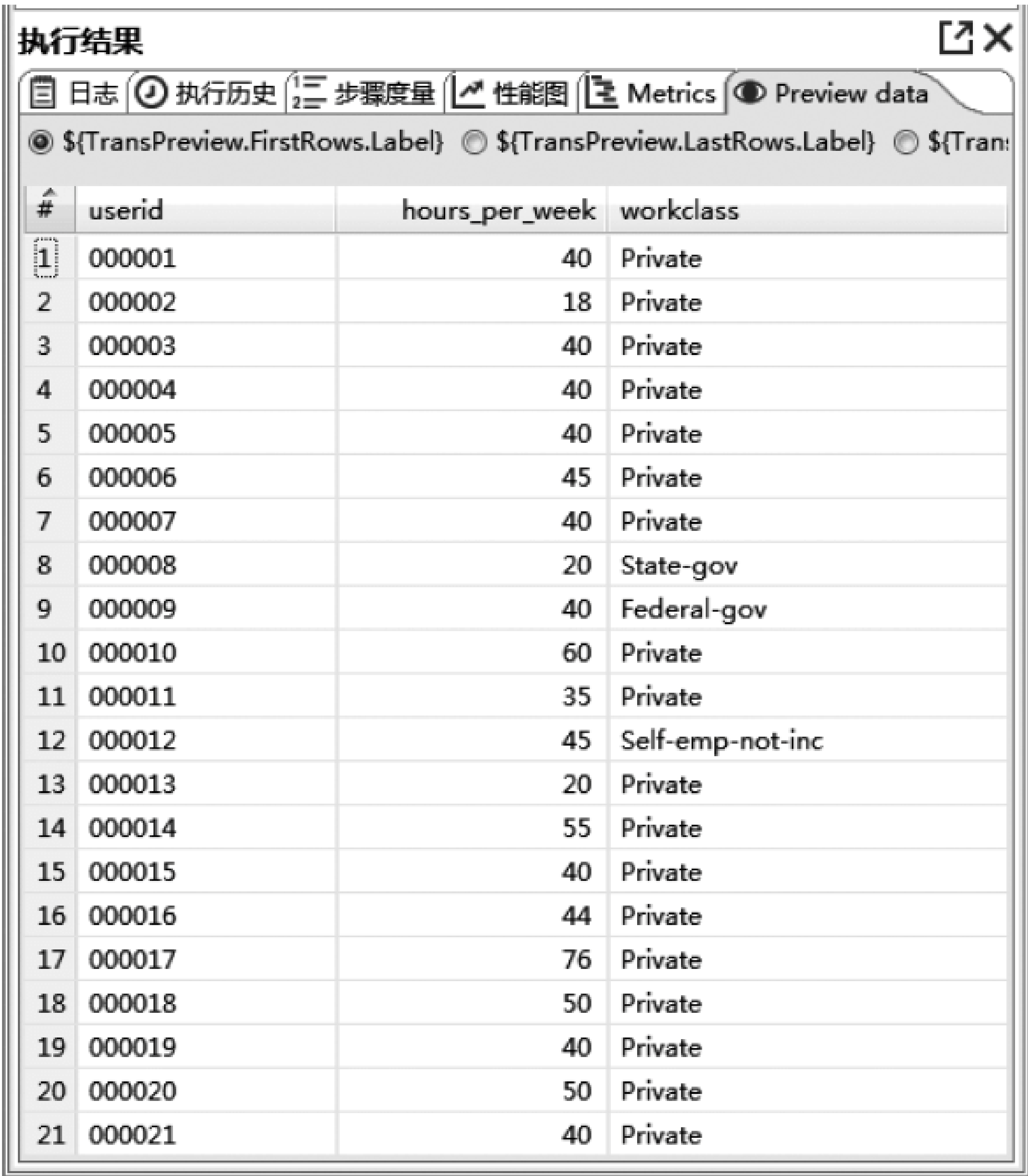
从图 5-79 中执行结果窗口的“步骤度量”选项卡可以看出，“文本文件输入”控件输入 21 条数据并写入该控件；“过滤记录”控件读取“文本文件输入”控件中的 21 条数据并写入该控件；“空操作(什么也不做)”控件读取符合过滤要求的 15 条数据并写入该控件；“空操作(什么也不做)2”控件读取不符合过滤要求的 6 条数据并写入该控件；“替换 NULL 值”控件读取“空操作”控件中的 15 条数据进行空值替换操作并写入该控件；“合并记录”控件读取“替换 NULL 值”控件和“空操作(什么也不做)2”控件共 21 条数据并写入该控件；“替换 NULL 值 2”控件读取“合并记录”中的 21 条数据进行空值替换操作并写入该控件；“字段选择”控件读取“替换 NULL 值 2”控件中的 21 条数据并写入该控件。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	文本文件输入	0	0	21	21	0	1	0	0	已完成
2	过滤记录	0	21	21	0	0	0	0	0	已完成
3	空操作 (什么也不做)	0	15	15	0	0	0	0	0	已完成
4	空操作 (什么也不做) 2	0	6	6	0	0	0	0	0	已完成
5	替换NULL值	0	15	15	0	0	0	0	0	已完成
6	合并记录	0	21	21	0	0	0	0	0	已完成
7	替换NULL值 2	0	21	21	0	0	0	0	0	已完成
8	字段选择	0	21	21	0	0	0	0	0	已完成

图 5-79　运行转换 fill_missing_value

单击图 5-79 中的“字段选择”控件，再单击执行结果窗口的 Preview data 选项卡，查看是否填充了文件 people_survey. txt 中的缺失值，具体如图 5-80 所示。



#	userid	hours_per_week	workclass
1	000001	40	Private
2	000002	18	Private
3	000003	40	Private
4	000004	40	Private
5	000005	40	Private
6	000006	45	Private
7	000007	40	Private
8	000008	20	State-gov
9	000009	40	Federal-gov
10	000010	60	Private
11	000011	35	Private
12	000012	45	Self-emp-not-inc
13	000013	20	Private
14	000014	55	Private
15	000015	40	Private
16	000016	44	Private
17	000017	76	Private
18	000018	50	Private
19	000019	40	Private
20	000020	50	Private
21	000021	40	Private

图 5-80　查看是否填充了文件 people_survey. txt 中的缺失值

从图 5-80 中可以看出，文件 people_survey. txt 中不存在缺失值数据了，说明通过 Kettle 工具实现了缺失值的填充。

注意：本小节操作只是将文件 people_survey. txt 的数据读取到 Kettle 中进行填充缺失值处理，并不会改变文件 people_survey. txt 的原始数据，如需保存处理后的数据，须添加相关输出控件。

5.3 异常值

异常值是指样本中的个别值，其数值明显偏离它所属样本的其余观测值，这些数值是不合理的或错误的。本节将针对出现异常值的原因、检测异常值、删除包含异常值的记录以及修补异常值进行讲解。

5.3.1 出现异常值的原因

遇到异常值时，处理它们的理想方法是先找出出现这些异常值的原因。下面通过一张表介绍常见的出现异常值的原因，具体见表 5-2。

表 5-2 常见的出现异常值的原因

异常值的类型	相关说明
数据输入错误	人为错误(如数据收集、记录或输入过程中引起的错误)可能导致数据异常。例如，客户的年收入为 100000 元。无意中，数据输入操作员增加了一个零，变成 1000000 元，该值即异常值
测量误差	测量误差是由于测量仪器发生故障导致的，该类异常值最常见。例如，有 10 台称重机，其中 9 台是完好的，1 台是有缺陷的。若使用有缺陷的称重机测量质量，则会高于/低于该组中的其他称重机测量的质量，该质量值即异常值
故意异常值	该类异常值通常出现在涉及敏感数据的自我报告的度量中。例如，青少年通常会报告他们消耗的酒精量，然而只有一小部分青少年会报告实际消耗的酒精量，这里实际消耗的酒精量看起来可能像异常值
数据处理错误	进行数据挖掘时，我们会从多个数据源中抽取数据，由于某些操作或抽取错误，可能会导致数据集中出现异常值
采样错误	当测量跳水运动员的身高时，也测量了篮球运动员的身高，并把篮球运动员的身高记入样本中，这样会导致样本中的数据集出现异常值
自然异常值	如果异常值不是人为原因造成的，就有可能是自然异常值。例如，一家知名的保险公司的前 50 名财务顾问的表现能力都强于其他人群，关于表现能力强弱的值即自然异常值

5.3.2 检测异常值

假设数据集中的大多数实例都是在正常的前提下，异常值的检测方法通常分为三大类，即无监督式异常值的检测、监督式异常值的检测以及半监督式异常值的检测，具体介绍如下。

- 无监督式异常值的检测，通过寻找与其他数据最不匹配的实例检测出未标记测试数据的异常。

- 监督式异常值的检测，需要一个已经被标记“正常”与“异常”的数据集，并涉及训练分类器，用来区分正常值和异常值。
- 半监督式异常值的检测，根据一个给定的正常训练数据集创建一个表示正常行为的模型，将检测的偏离正常行为的对象视为异常值。

一般异常值的检测方法包含基于统计的方法、基于聚类的方法以及一些专门检测异常值的方法等异常值检测的方法。下面针对这些异常值检测的方法进行详细讲解。

1. 简单统计方法

对属性值进行一个描述性的统计，从而查看哪些值是不合理的。例如，对年龄这个属性进行规约：年龄的区间为[0：150]，如果样本中的年龄值不在该区间范围内，则表示该样本的年龄属性属于异常值。

2. 3σ 准则

3σ 原则又称为拉依达原则，它是指假设一组检测数据只含有随机误差，对其进行计算处理得到标准偏差，按一定概率确定一个区间，凡是超过这个区间的误差，都是粗大误差，相应的数据应予以剔除。

在正态分布概率公式中，σ 表示标准差，μ 表示平均数，$f(x)$表示正态分数函数，具体如下：

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]\text{（正态分布公式）}$$

下面通过一张图描述正态分布函数，具体如图 5-81 所示。

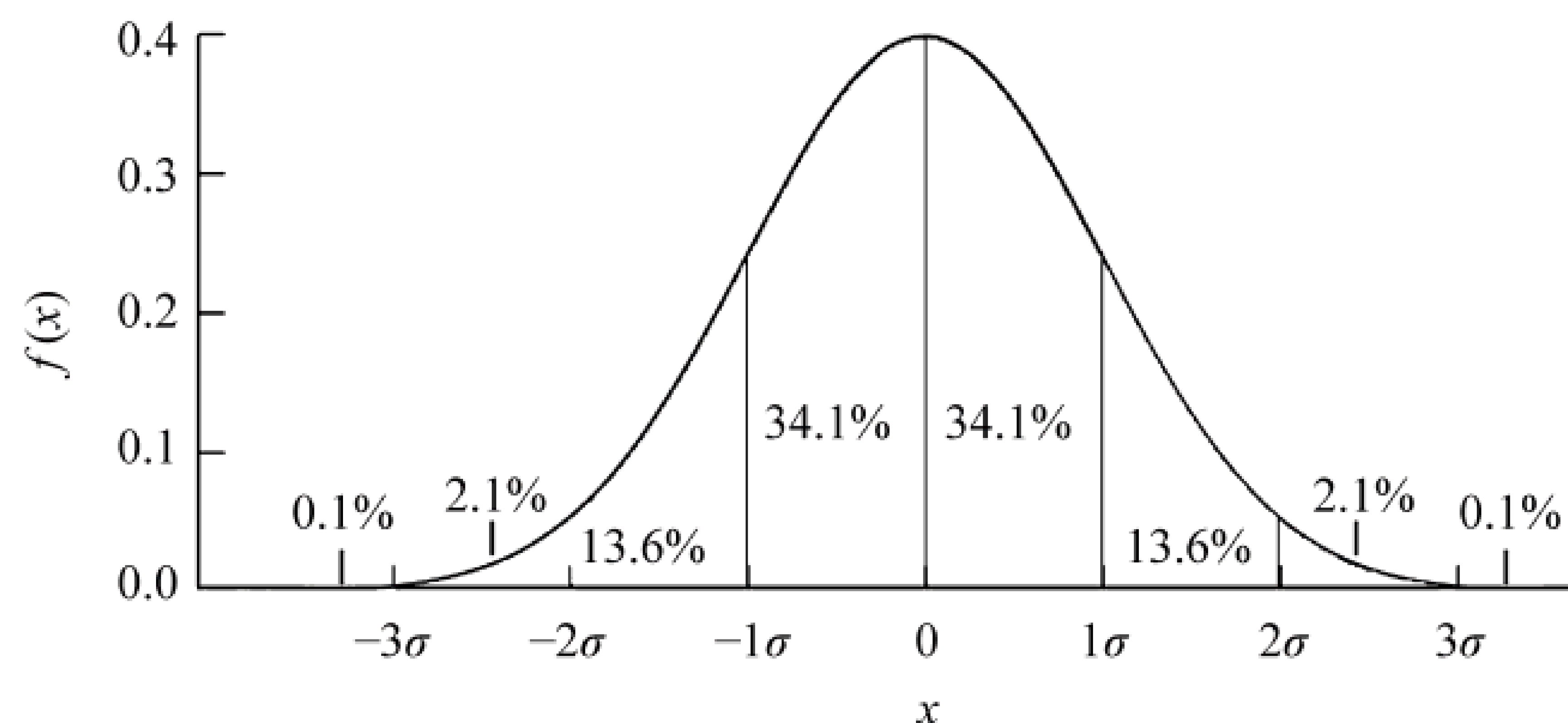


图 5-81　正态分布函数图

从图 5-81 可以看出，3σ 原则在各个区间所占的概率如下所示。

（1）数值分布在$(\mu-\sigma,\mu+\sigma)$中的概率为 0.682。

（2）数值分布在$(\mu-2\sigma,\mu+2\sigma)$中的概率为 0.954。

（3）数值分布在$(\mu-3\sigma,\mu+3\sigma)$中的概率约为 0.997。

由此可知，数值几乎全部集中在$(\mu-3\sigma,\mu+3\sigma)$区间，超出这个范围的可能性仅占不到 0.3%。所以，凡是误差超过这个区间的就属于异常值，应予以剔除。

3. 箱形图

箱形图又称为箱线图，是一种用于显示一组数据分散情况的统计图。在箱形图中，异常

值通常被定义为小于 QL－1.5IQR 或大于 QU＋1.5IQR 的值。其中，

(1) QL 称为下四分位数，表示全部观察中四分之一的数据取值比它小。

(2) QU 称为上四分位数，表示全部观察值中有四分之一的数据取值比它大。

(3) IQR 称为四分位数间距，是上四分位数 QU 与下四分位数 QL 之差，其间包含了全部观察值的一半。

离散点表示的是异常值，上界表示除异常值以外数据中的最大值；下界表示除异常值以外数据中的最小值，具体如图 5-82 所示。

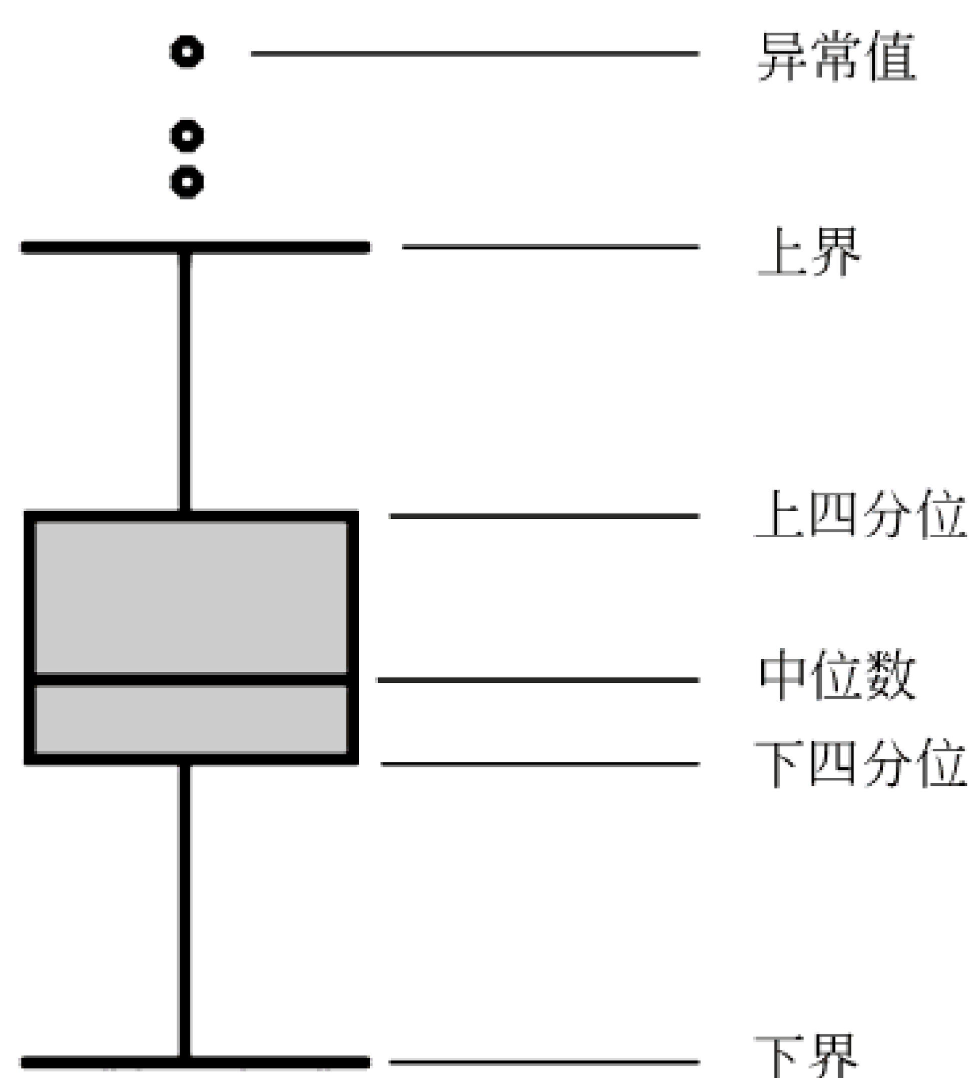


图 5-82　箱形图结构示意图

图 5-82 中的箱形图是根据实际数据进行绘制的，对数据没有任何要求(如 3σ 原则要求数据服从正态分布或近似正态分布)。箱形图判断异常值的标准是以四分位数和四分位距为基础的。

4. 基于邻近的模型

基于邻近的模型是评估值与其他值孤立情况的模型，该模型主要分为三类，即聚类分析、基于密度的分析以及最近邻分析，具体介绍如下。

(1) 在聚类的分析中，首先建立一个数据模型，异常是那些同模型不能完美拟合的对象；如果模型是簇的集合，则异常是不显著属于任何簇的对象；使用回归模型时，异常是相对远离预测值的对象。

(2) 基于密度的分析中，仅当一个点的局部密度显著低于它的大部分近邻时，才将其分类为离群点。

(3) 在最近邻近度的分析中，有两种不同的分析策略：第一种策略是采用给定邻域半径，依据点的邻域中包含的对象多少判定离群点，如果一个点的邻域内包含的对象少于整个数据集的一定比例，则标识它为离群点；第二种策略是利用 k 最近邻距离的大小判定离群点，若 k 太小(如 1)，则少量的邻近离群点可能导致较低的离群程度；若 k 太大，则点数少于 k 的簇中所有的对象可能都成了离群点。

5.3.3　删除包含异常值的记录

不少人在处理异常值时，习惯于使用简单粗暴的删除方法，但是这并不适用于处理所有异常值，若是通过删除的方法处理异常值，有时会使你错过真正的规律。如果想删除数据，一定需要有合理的理由。当出现异常值时，需要先清楚是什么原因导致出现异常值，然后考虑如何处理。

如果异常值是由于输入或测量数据不正确而造成，则很明显就能看出这一类数据是错的。例如，假设有一个关于体重的数据集，其中女性的体重记录为 19 斤，通常这样的体重对于一个女性来说是不正常的，她的真实体重可能是 119 斤或 190 斤等，由于我们无法确定是哪个值，因此这一类数据可以直接删除。

极端异常值会对平均值产生很大影响，但不会影响中位数。因此，如果只计算中位数，则可以包含异常值。如果异常值过于极端而不可信(例如，可能由于测量误差)，则应该将其

排除；如果异常值是合理的，则需要分析是否有异常值数据。如果这两种类型的数据分析得出的结果一致，则可以删除该异常值。

下面介绍几种不宜删除异常值的情况。

（1）通常，数据中出现的异常值较少。如果采集的数据中有超过 30％的异常值数据，就意味着需要进一步研究数据。

（2）如果异常值存在且代表了一种真实存在的现象，那就不可随意删除。例如，调查 100 个村的胃癌发病率，可能确实有个别村庄的发病率远远高于其他村，这时就不能随意删除，而是要把这些异常点纳入，重新拟合模型，研究其规律。

（3）分析数据的结果至关重要，因此即使很小的变化，也会很重要。例如，可以更好地放弃关于人们最喜欢电视节目的异常值，而不是放弃关于飞机封条失效的温度。

假设有一份记录一天中不同时间的温度数据文件 temperature. txt，其中包含时间和温度（摄氏度）两个字段，具体内容如图 5-83 所示（展示部分数据）。

```
00:00	84
00:30	84
01:00	84
01:30	84
02:00	84
02:30	84
03:00	84
03:30	84
04:00	84
04:30	84
05:00	84
05:30	84
06:00	84
06:30	137
07:00	84
07:30	84
08:00	84
08:30	86
```

图 5-83　文件 temperature. txt 的内容

在图 5-83 中，温度数据可通过箱形图的四分位数计算出 5 个统计量，即下限是 76. 5、下四分位是 84、中位数是 84、上四分位是 89、上限是 96. 5，因此可以确定非异常值的取值范围是[76. 5，96. 5]。

下面通过 Kettle 工具分步骤讲解如何去除文件 temperature. txt 中的异常值，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 delete_anomalous _value，并添加“文本文件输入”控件、“过滤记录”控件、“空操作（什么也不做）”控件以及 Hop 跳连接线，具体效果如图 5-84 所示。

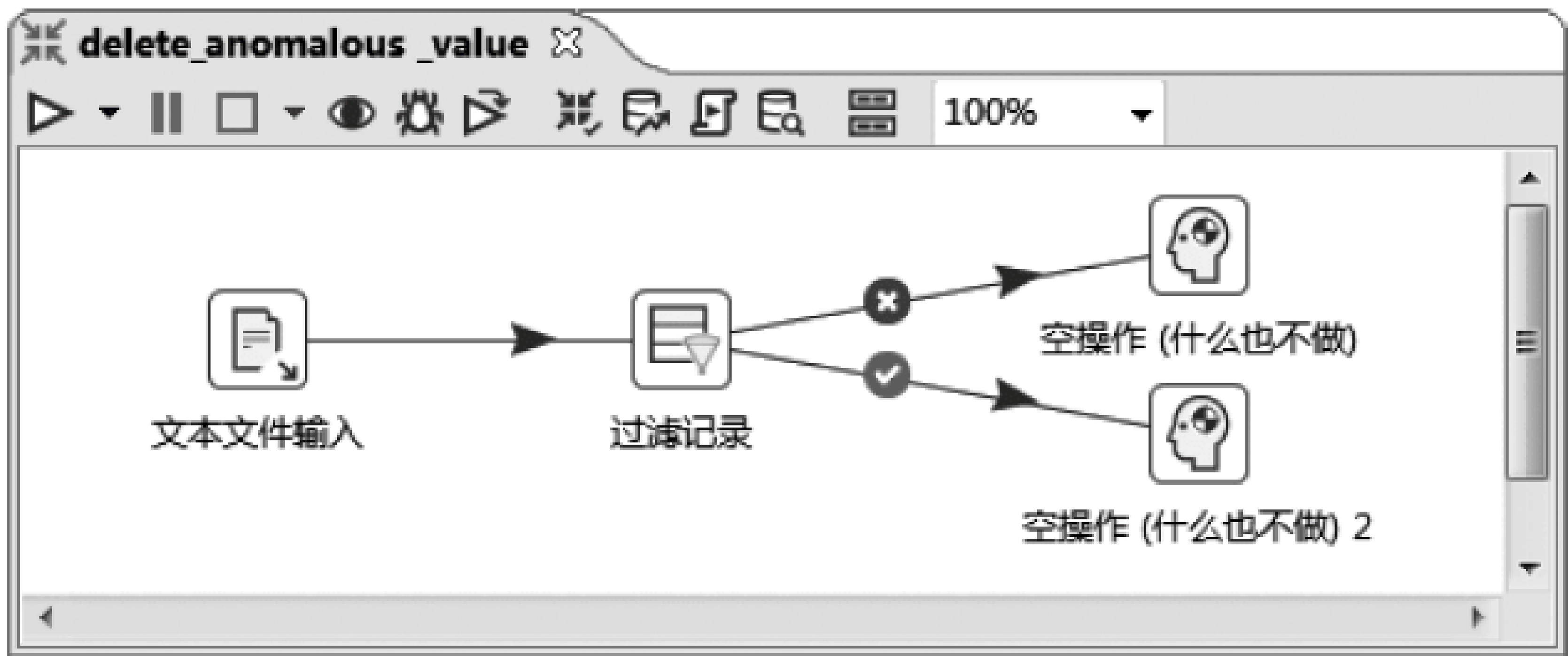


图 5-84　创建转换 delete_anomalous _value

2. 配置“文本文件输入”控件

双击图 5-84 中的“文本文件输入”控件，进入“文本文件输入”界面，如图 5-85 所示。

在图 5-85 中先单击“浏览”按钮，选择要去除异常值的文件 temperature. txt，然后单击“增加”按钮，将要去除异常值的文件 temperature. txt 添加到“文本文件输入”控件中，具体



图 5-85 “文本文件输入”界面

效果如图 5-86 所示。



图 5-86 添加文件 temperature.txt 到“文本文件输入”控件

在图 5-86 中单击“内容”选项卡，切换到“内容”选项卡界面，如图 5-87 所示。

在图 5-87 中清除分隔符处的默认分隔符“;”，单击 Insert TAB 按钮，在分隔符处插入一个制表符；取消勾选“头部”复选框，若不取消，则在进行数据抽取操作时会排除文件第一行的数据。“内容”选项卡的配置如图 5-88 所示。

在图 5-88 中单击“字段”选项卡，切换到“字段”选项卡界面，如图 5-89 所示。

在图 5-89 中根据文件 temperature.txt 的内容添加对应的字段名称，并指定数据类型，这里需要注意，制表符可看作是由多个空格组成，因此在“去除空字符串方式”列时，所添加的字段都应选择“不去掉空格”，否则在抽取数据操作时会把制表符当作空格去除，而不能把制表符作为分隔符实现文本文件内容的分隔，如图 5-90 所示。



图 5-87　“内容”选项卡界面



图 5-88　“内容”选项卡的配置



图 5-89　“字段”选项卡界面



图 5-90　添加文件 temperature.txt 中的字段

在图 5-90 中单击“预览记录”按钮，查看文件 temperature.txt 的数据是否成功抽取到文本文件输入流中，具体效果如图 5-91 所示。

从图 5-91 中可以看出，文件 temperature.txt 的数据已经成功抽取到文本文件输入流中(注：这里只截取一部分数据进行展示)。单击“关闭”→“确定”按钮，完成“文本文件输入”控件的配置。

3. 配置“过滤记录”控件

双击图 5-84 中的“过滤记录”控件，进入“过滤记录”界面，如图 5-92 所示。

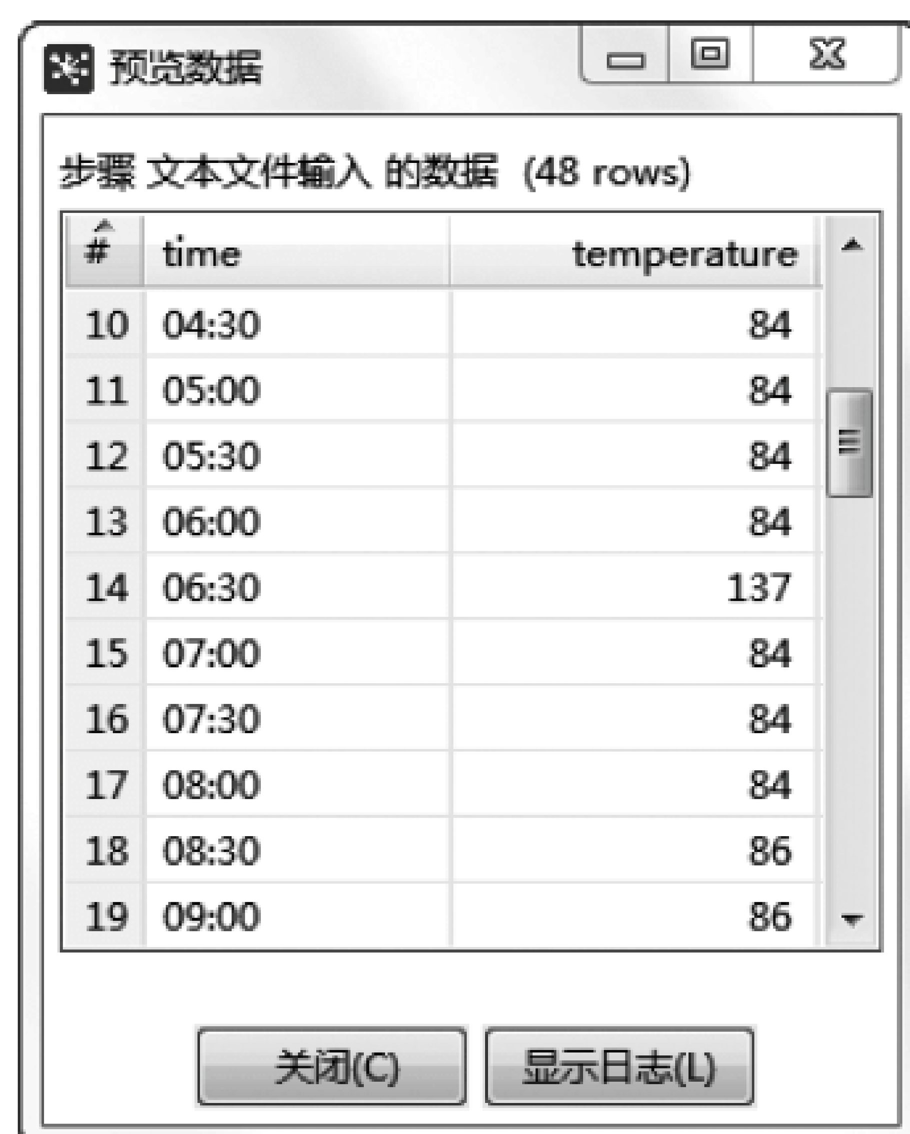


图 5-91　预览数据



图 5-92　“过滤记录”界面

在图 5-92 中的“条件”处设置过滤的条件，由于文件 temperature.txt 中 time 为 6：30 的温度是 137 摄氏度，不在非异常值的范围内，因此属于异常值，应该将过滤字段设置为 temperature、过滤值设置为 137，具体如图 5-93 所示。

在图 5-93 中的“发送 true 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)2”，将异常值放在“空操作(什么也不做)2”控件中；在“发送 false 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)”，将非异常值放在“空操作(什么也不做)”控件中，具体如图 5-94 所示。



图 5-93　设置过滤条件

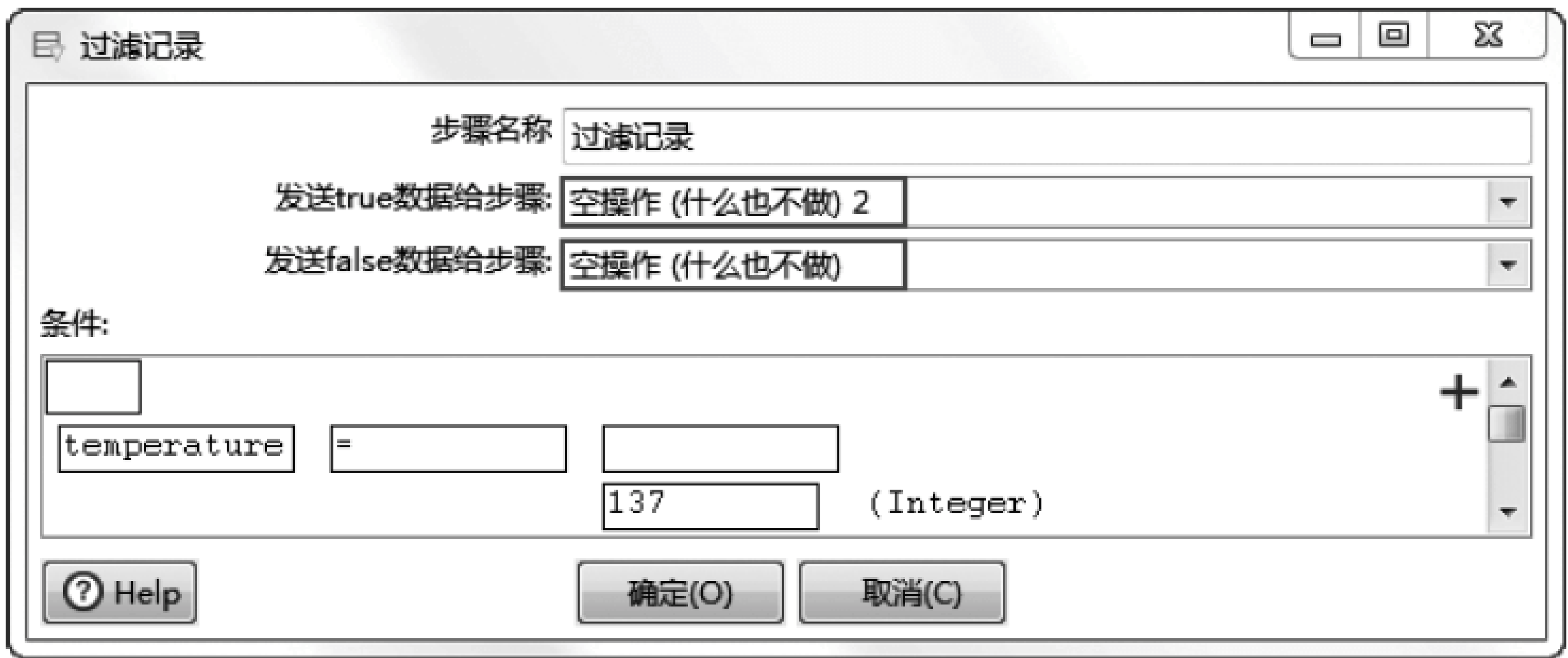
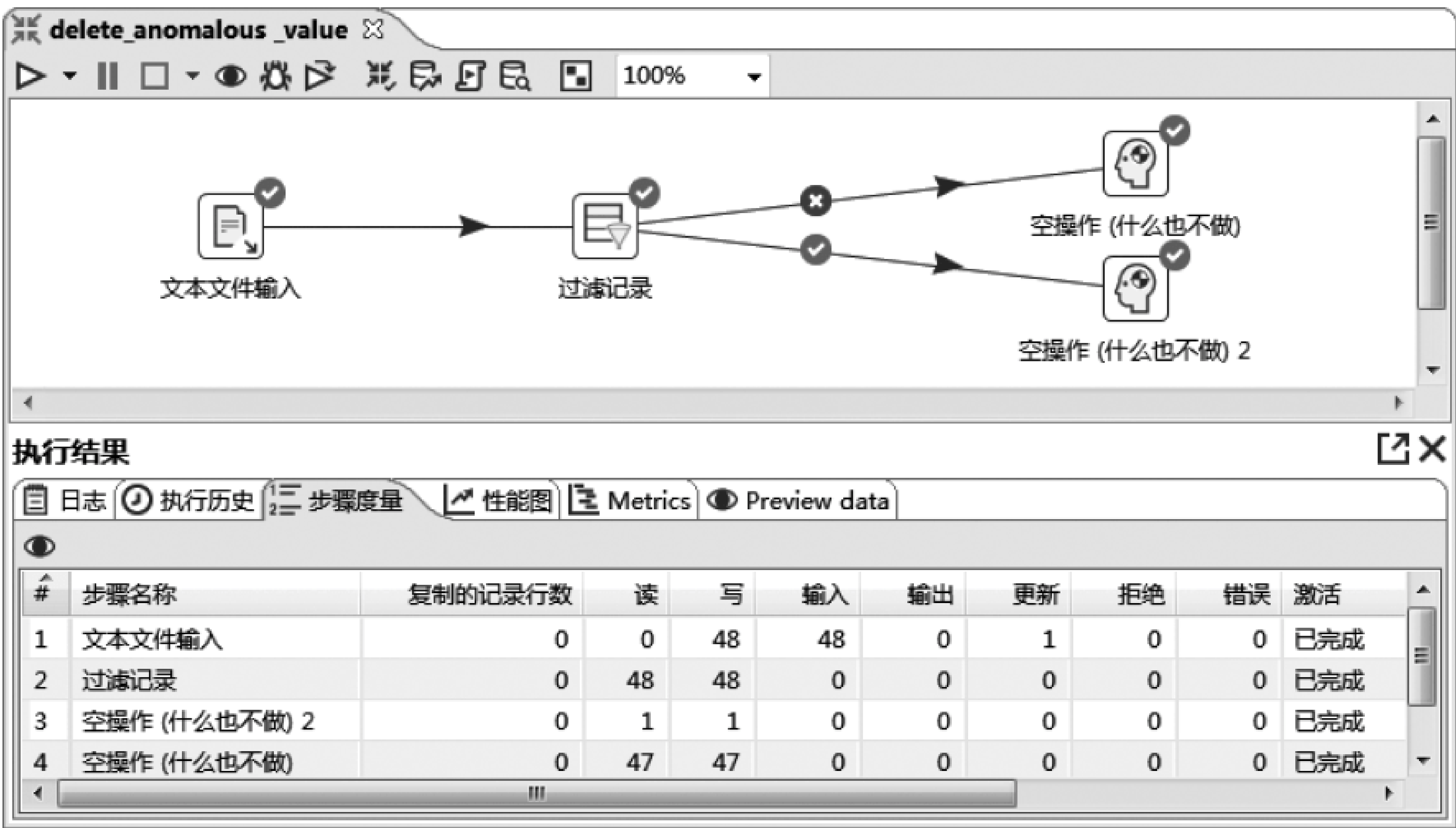


图 5-94　配置发送 true/false 数据给相关步骤

在图 5-94 中单击“确定”按钮，完成“过滤记录”控件的配置。

4．运行转换 delete_anomalous_value

单击转换工作区顶部的▷按钮，运行创建的转换 delete_anomalous_value，实现去除文件 temperature. txt 中的异常值，具体如图 5-95 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	文本文件输入	0	0	48	48	0	1	0	0	已完成
2	过滤记录	0	48	48	0	0	0	0	0	已完成
3	空操作 (什么也不做) 2	0	1	1	0	0	0	0	0	已完成
4	空操作 (什么也不做)	0	47	47	0	0	0	0	0	已完成

图 5-95　运行转换 delete_anomalous_value

从图 5-95 中执行结果窗口的“步骤度量”选项卡可以看出,“文本文件输入”控件执行 48 行写操作、48 行输入操作以及 1 行更新操作;“过滤记录”控件执行读、写操作各 48 行;“空操作(什么也不做)2”控件执行读、写操作各 1 行;“空操作(什么也不做)”控件执行读、写操作各 47 行。也就是说,文件 temperature. txt 中有 1 行是异常值,其余 47 行均为正常值。

单击图 5-95 中的“空操作”控件,再单击执行结果的 Preview data 选项卡,查看是否去除了异常值,具体如图 5-96 所示。

#	time	temperature	#	time	temperature
1	00:00	84	25	12:30	90
2	00:30	84	26	13:00	88
3	01:00	84	27	13:30	88
4	01:30	84	28	14:00	90
5	02:00	84	29	14:30	90
6	02:30	84	30	15:00	90
7	03:00	84	31	15:30	90
8	03:30	84	32	16:00	90
9	04:00	84	33	16:30	90
10	04:30	84	34	17:00	90
11	05:00	84	35	17:30	90
12	05:30	84	36	18:00	90
13	06:00	84	37	18:30	86
14	07:00	84	38	19:00	86
15	07:30	84	39	19:30	84
16	08:00	84	40	20:00	84
17	08:30	86	41	20:30	84
18	09:00	86	42	21:00	84
19	09:30	86	43	21:30	84
20	10:00	86	44	22:00	84
21	10:30	88	45	22:30	84
22	11:00	88	46	23:00	84
23	11:30	88	47	23:30	82
24	12:00	90			

图 5-96　查看是否去除 temperature. txt 文件的异常值

从图 5-96 中可以看出,文件 temperature. txt 中不存在异常数据了,说明我们通过 Kettle 工具实现了去除异常值的功能。

5.3.4　修补异常值

通过直接删除的方式处理异常值虽然是最直接的方法,但是会减少数据样本,因此,在数据集小的情况下减少数据样本会对结果产生影响;在含有较多异常值的数据集中,大量删除异常值也会对结果产生影响。因此,在异常值没有可研究性的情况下,应该对这些异常值进行修补处理。

修补异常值的方式主要有两种,即修改异常值和替换异常值,这两种方式的具体介绍如下。

1. 修改异常值

修改异常值有两种策略:一是利用数据集中的代表性属性,如众数或均值等,或是定义一个数据替代异常值;二是通过回归模型、决策树模型、贝叶斯定理等预测异常值,并利用最邻近值替代异常值。前者是人为替代异常值,不能完全代表异常值本身的真实含义,后者是

将异常数据对应的变量当作目标变量,把其他的输入变量作为自变量,为每个需要进行异常值赋值的字段分别建立预测模型,从而利用最邻近值替代异常值,可以近似代表异常值本身的含义。

2. 替换异常值

替换异常值是将异常值替换成缺失值,然后按照缺失值数据处理的方法进行处理。

假设有一份 500 人的身高调查数据表 interpolation_data,其中包括 id、Gender 和 Height 字段,具体数据内容如图 5-97 所示(注:这里只截取了部分人的数据)。

id	Gender	Height
1	Male	174
2	Male	189
3	Female	185
4	Female	195
5	Male	149
6	Male	189
7	Male	147
8	Male	154
9	Male	174
10	Female	169
11	Male	195
12	Female	159
13	Female	192
14	Male	155
15	Male	260
16	Female	153
17	Female	157
18	Male	140
19	Male	144
20	Male	172

图 5-97　数据表 interpolation_data

图 5-97 中的身高数据可通过箱形图的四分位数计算得出 5 个统计量,即下限是 114、下四分位是 156、中位数是 170、上四分位是 184、上限是 226,因此可以确定非异常值的取值范围是[114,226]。

下面通过 Kettle 工具分步骤讲解如何替换和修改数据表 interpolation_data 中的异常值,具体步骤如下。

1. 打开 Kettle 工具,创建转换

使用 Kettle 工具创建转换 fill_unusual_value,并添加"表输入"控件、"过滤记录"控件、"空操作(什么也不做)"控件、"设置值为 NULL"控件、"合并记录"控件、"替换 NULL 值"控件、"字段选择"控件以及 Hop 跳连接线,具体效果如图 5-98 所示。

2. 配置"表输入"控件

双击图 5-98 中的"表输入"控件,进入"表输入"界面,如图 5-99 所示。

在图 5-99 中单击"新建"按钮,配置数据库连接,配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 5-100 所示。

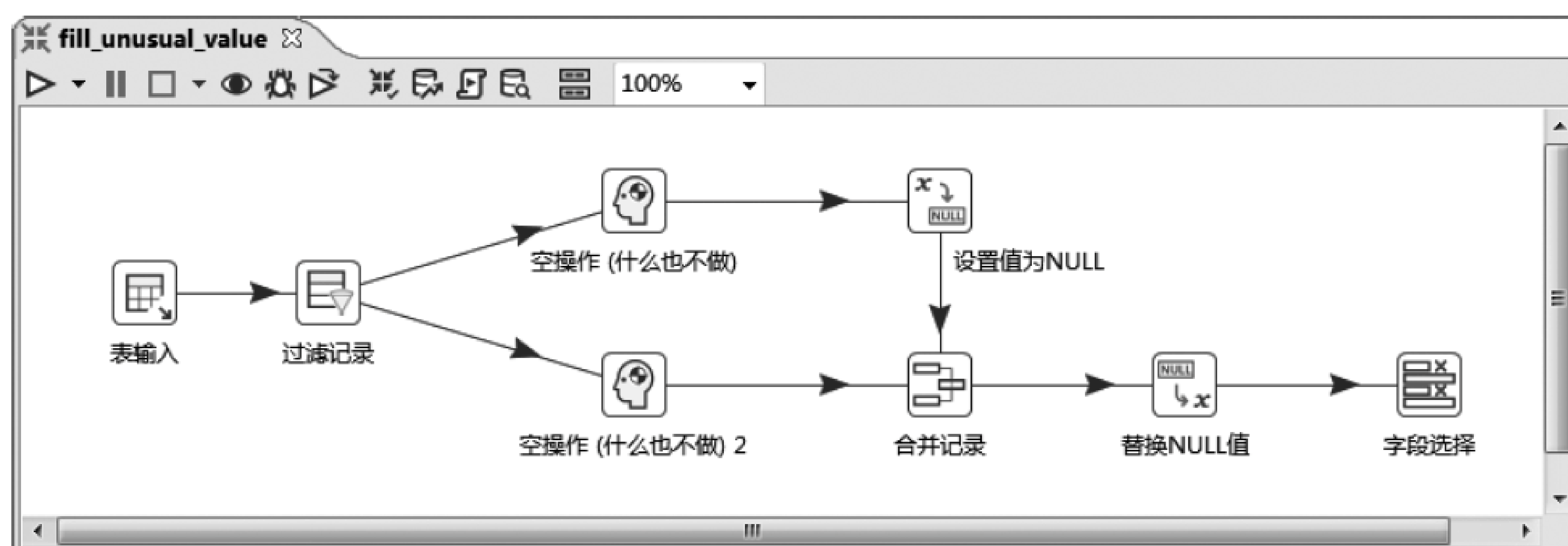


图 5-98 创建转换 fill_unusual_value



图 5-99 “表输入”界面

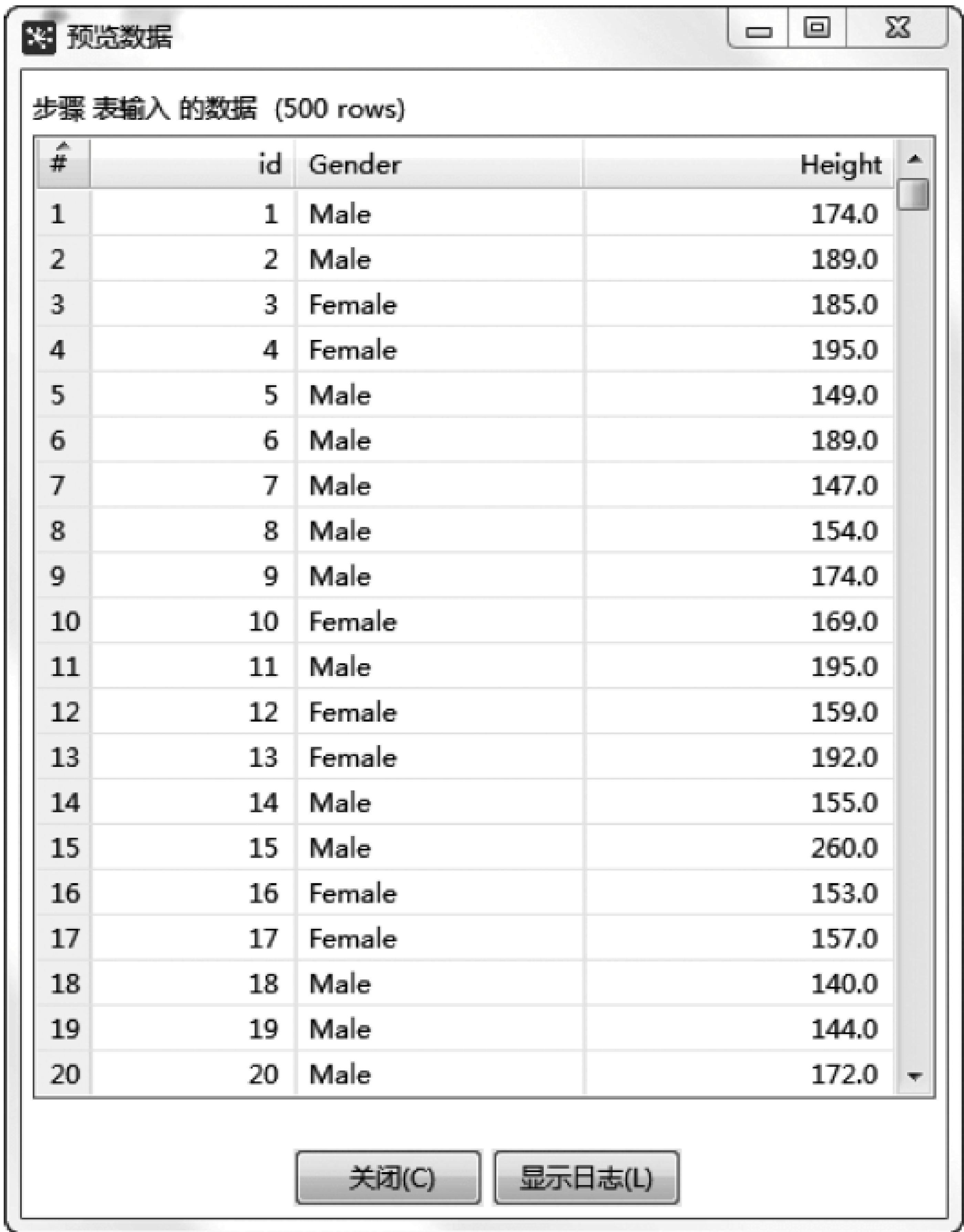


图 5-100 MySQL 数据库连接的配置

在图 5-99 的 SQL 框中编写查询数据表 interpolation_data 的 SQL 语句，然后单击“预览”按钮，查看数据表 interpolation_data 的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 5-101 和图 5-102 所示。



图 5-101　编写 SQL 语句


预览数据

步骤 表输入 的数据 (500 rows)

#	id	Gender	Height
1	1	Male	174.0
2	2	Male	189.0
3	3	Female	185.0
4	4	Female	195.0
5	5	Male	149.0
6	6	Male	189.0
7	7	Male	147.0
8	8	Male	154.0
9	9	Male	174.0
10	10	Female	169.0
11	11	Male	195.0
12	12	Female	159.0
13	13	Female	192.0
14	14	Male	155.0
15	15	Male	260.0
16	16	Female	153.0
17	17	Female	157.0
18	18	Male	140.0
19	19	Male	144.0
20	20	Male	172.0

关闭(C)　显示日志(L)

图 5-102　预览数据

从图 5-102 中可以看出，数据表 interpolation_data 的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“过滤记录”控件

双击图 5-98 中的“过滤记录”控件，进入“过滤记录”界面，如图 5-103 所示。



图 5-103 “过滤记录”界面

在图 5-103 中的“条件”处设置过滤的条件，即设置 Height 字段的取值范围([114，226])，从而判断数据表中的每个数据是否为异常值。若在非异常值的取值范围内，则是非异常值，否则是异常值。过滤条件的设置如图 5-104 所示。



图 5-104 过滤条件的设置

在图 5-104 中“发送 true 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)2”，将非异常值放在“空操作(什么也不做)2”控件中；在“发送 false 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)”，将异常值放在“空操作(什么也不做)”控件中，具体如图 5-105 所示。

在图 5-105 中单击“确定”按钮，完成“过滤记录”控件的配置。

4. 预览“空操作(什么也不做)”控件中的数据

选中图 5-98 中的“空操作(什么也不做)”控件，然后单击转换工作区顶部的 👁 按钮，预览“空操作(什么也不做)”控件中的数据，具体如图 5-106 所示。

从图 5-106 中可以看出，id 为 15 的这条数据，Height 字段为 260.0，260.0 不在非异常值范围[114，226]内，因此该条数据为异常数据。



图 5-105　配置发送 true/false 数据给相关步骤



图 5-106　预览“空操作(什么也不做)”控件中的数据

5. 配置“设置值为 NULL”控件

双击图 5-98 中的“设置值为 NULL”控件，进入“设置值为 NULL”界面，如图 5-107 所示。



图 5-107　“设置值为 NULL”界面

在图 5-107 中的“字段”处添加要设为 NULL 值的字段名称和需要转换成 NULL 的值，具体如图 5-108 所示。

在图 5-108 中单击“确定”按钮，完成“设置值为 NULL”控件的配置。

6. 配置“合并记录”控件

双击图 5-98 中的“合并记录”控件，进入“合并行(比较)”界面，如图 5-109 所示。



图 5-108 添加要设为 NULL 值的字段名称和需要转换成 NULL 的值

在图 5-109 中“旧数据源：”后的下拉列表中选择“设置为 NULL 值”，在“新数据源：”后的下拉列表中选择“空操作(什么也不做)2”；在“匹配的关键字：”处添加关键字段，即 id，具体如图 5-110 所示。



图 5-109 “合并行(比较)”界面



图 5-110 配置“合并记录”控件

“合并记录”控件主要是将两个数据源(旧数据源、新数据源)进行合并，标志字段主要是将每条数据进行标记，新数据源的数据会标记为 new，旧数据源的数据会标记为 deleted，若新、旧数据源中存在相同的关键字段设置的数据，则两个数据源进行合并后只会保存从新数据源中获取的数据，并以 identical 进行标记。

在图 5-110 中单击“确定”按钮，完成“合并记录”控件的配置。

7. 配置“替换 NULL 值”控件

双击图 5-98 中的“替换 NULL 值”控件，进入“替换 NULL 值”界面，如图 5-111 所示。

在图 5-111 中勾选“选择字段”复选框，并在“字段”框添加字段 Height，值替换为 170(通过计算得到 499 人的平均身高值近似为 170，因此用 170 替换字段 Height 中的 NULL 值)，具体如图 5-112 所示。

在图 5-112 中单击“确定”按钮，完成“替换 NULL 值”控件的配置。

8. 配置“字段选择”控件

双击图 5-98 中的“字段选择”控件，进入“选择/改名值”界面，如图 5-113 所示。



图 5-111　"替换 NULL 值"界面

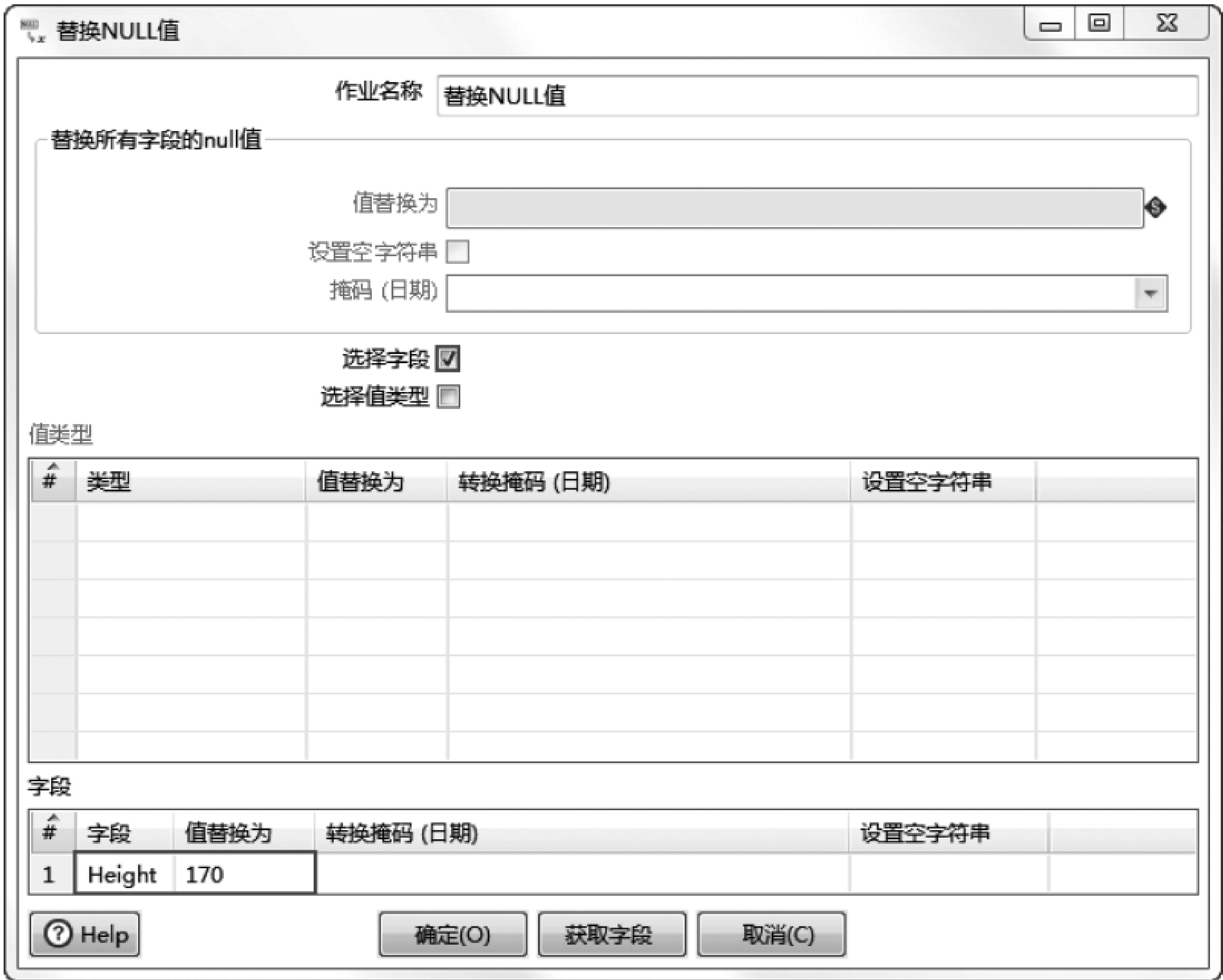


图 5-112　配置"替换 NULL 值"控件



图 5-113 “选择/改名值”界面

在图 5-113 的“移除”选项卡中添加要移除的字段名称，这里移除的是字段 flagfield，如图 5-114 所示。



图 5-114 添加要移除的字段

在图 5-114 中单击“确定”按钮，完成“字段选择”控件的配置。

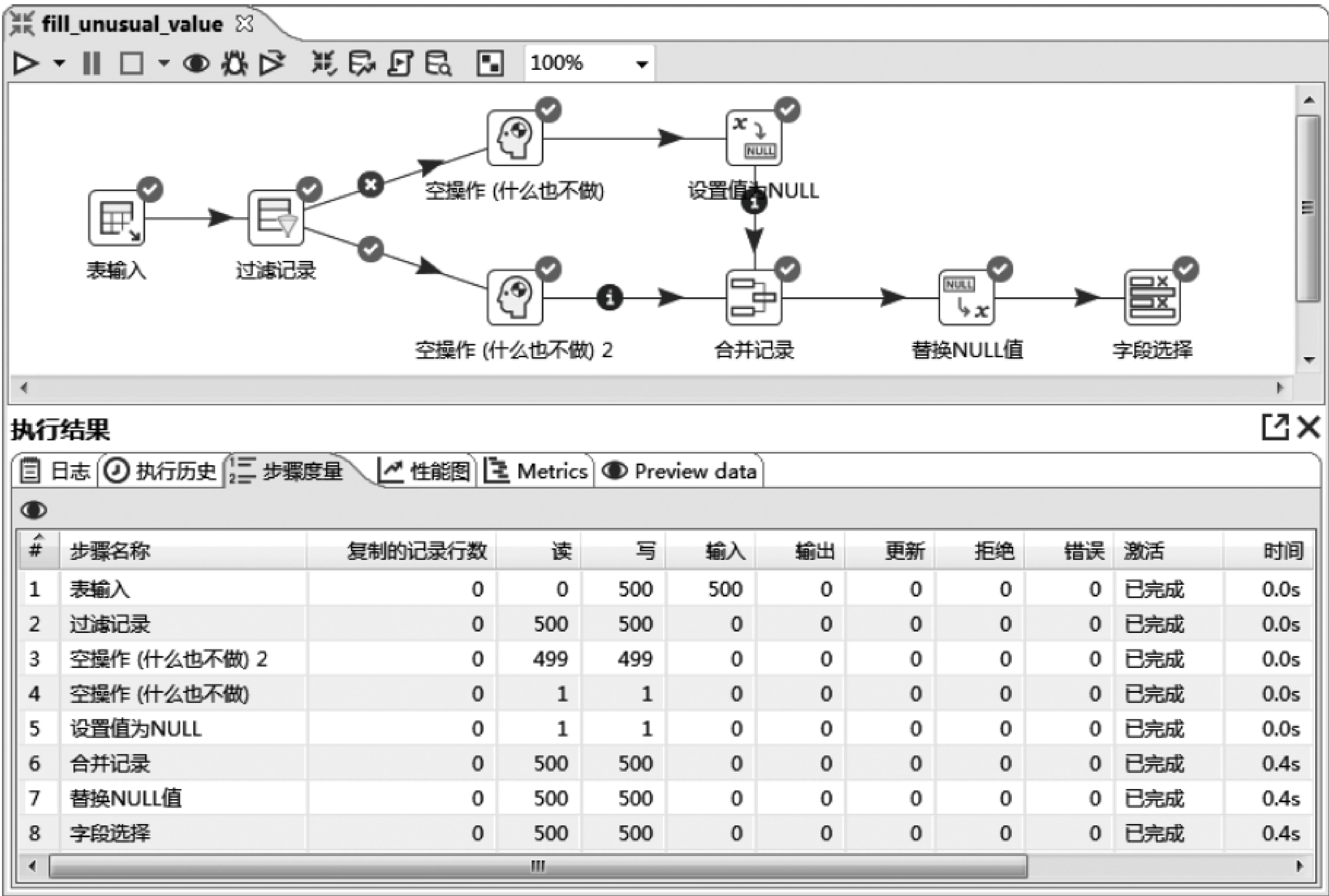
9．运行转换 fill_unusual_value

单击转换工作区顶部的 ▷ 按钮，运行创建的转换 fill_unusual_value，实现修改并替换数据表 interpolation_data 中的异常值的功能，具体如图 5-115 所示。

从图 5-115 中执行结果窗口的“步骤度量”选项卡可以看出，“表输入”控件输入 500 条数据并写入该控件；“过滤记录”控件读取“表输入”控件中的 500 条数据并写入该控件；“空操作(什么也不做)2”控件读取符合过滤要求的 499 条数据并写入该控件；“空操作(什么也不做)”控件读取不符合过滤要求的 1 条数据并写入该控件；“设置值为 NULL”控件读取“空操作(什么也不做)”控件中的 1 条数据并写入该控件；“合并记录”控件读取“设置值为 NULL”控件和“空操作(什么也不做)2”控件的数据共 500 条并写入该控件；“替换 NULL 值”控件读取“合并记录”控件的 500 条数据并写入该控件；“字段选择”控件读取“替换 NULL 值”控件的 500 条数据并写入该控件。

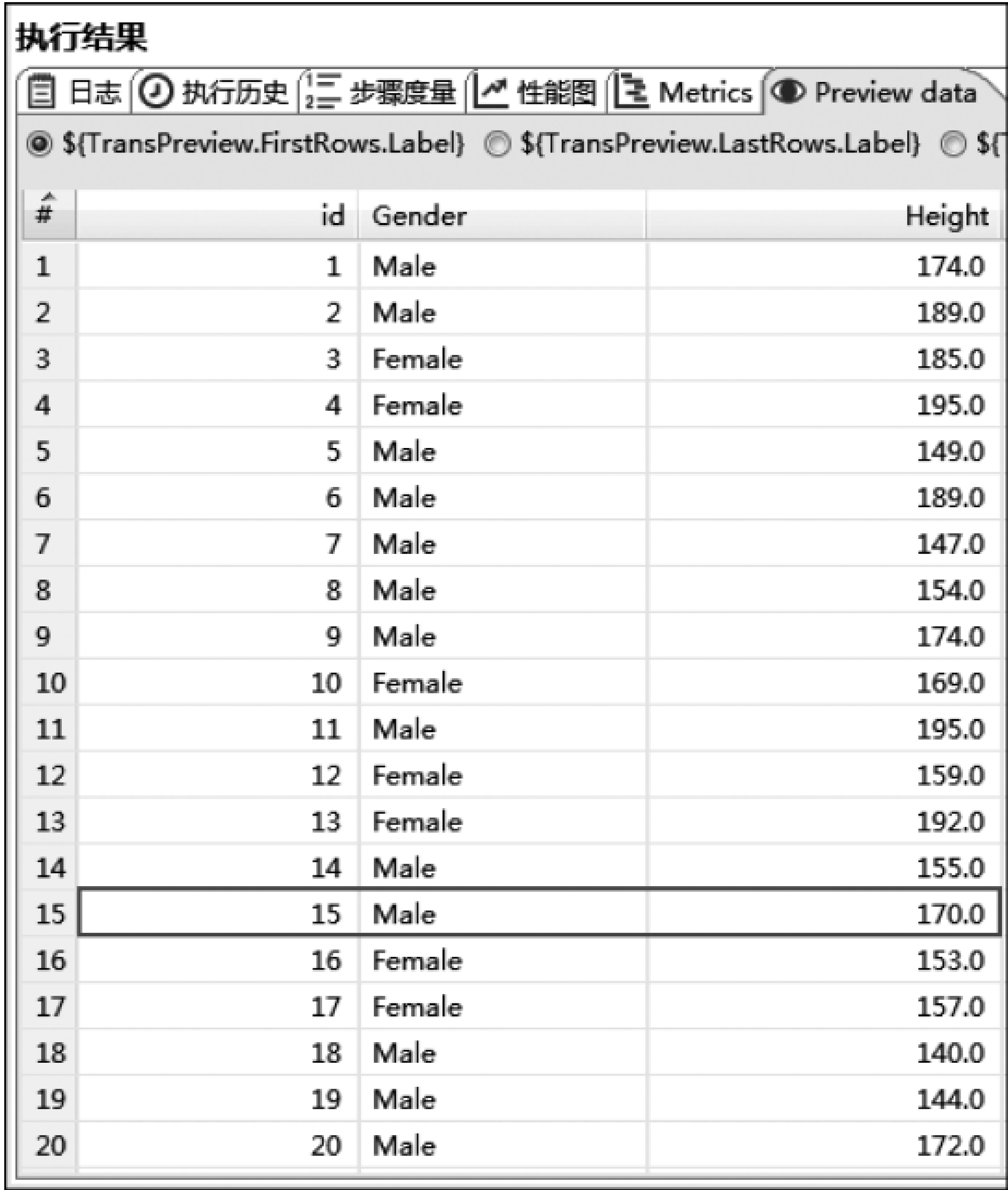
单击图 5-115 中的“字段选择”控件，再单击执行结果窗口的 Preview data 选项卡，查看是否修改并替换数据表 interpolation_data 中的异常值，具体如图 5-116 所示。

从图 5-116 中可以看出，数据表 interpolation_data 中不存在异常值数据，并把 id 为 15 的这条数据中的 Height 字段值替换成了 170，说明我们通过 Kettle 工具实现了修改并替换异常值的功能。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活	时间
1	表输入	0	0	500	500	0	0	0	0	已完成	0.0s
2	过滤记录	0	500	500	0	0	0	0	0	已完成	0.0s
3	空操作 (什么也不做) 2	0	499	499	0	0	0	0	0	已完成	0.0s
4	空操作 (什么也不做)	0	1	1	0	0	0	0	0	已完成	0.0s
5	设置值为NULL	0	1	1	0	0	0	0	0	已完成	0.0s
6	合并记录	0	500	500	0	0	0	0	0	已完成	0.4s
7	替换NULL值	0	500	500	0	0	0	0	0	已完成	0.4s
8	字段选择	0	500	500	0	0	0	0	0	已完成	0.4s

图 5-115　运行转换 fill_unusual_value



#	id	Gender	Height
1	1	Male	174.0
2	2	Male	189.0
3	3	Female	185.0
4	4	Female	195.0
5	5	Male	149.0
6	6	Male	189.0
7	7	Male	147.0
8	8	Male	154.0
9	9	Male	174.0
10	10	Female	169.0
11	11	Male	195.0
12	12	Female	159.0
13	13	Female	192.0
14	14	Male	155.0
15	15	Male	170.0
16	16	Female	153.0
17	17	Female	157.0
18	18	Male	140.0
19	19	Male	144.0
20	20	Male	172.0

图 5-116　查看是否修改并替换数据表 interpolation_data 中的异常值

5.4 数据检验

企业中的数据一般都要遵守预定义的业务规则，数据检验就是在数据清洗过程中，通过对数据项增加验证约束，实现对数据有效性的验证。本节将针对数据一致性以及规范化的校验进行讲解。

5.4.1 数据一致性处理

数据一致性是指在对一个副本数据进行更新的同时，必须确保也能够更新到其他的副本，否则不同的副本之间的数据将不再一致。例如，当你在某银行已存有 5000 元，接着又存了 1000 元，然后跑到另外一个地方游山玩水，需要在当地的银行把这 6000 元取出来，但是存钱的银行并没有及时将你存钱的信息传给当地的银行，所以当地银行还不知道你已经存了 1000 元进去，当地银行的工作人员告诉你账户余额只有 5000 元，这时你的同一账户的余额在不同地点就出现了不一致情况。

数据的一致性有 3 种类型，即强一致性、弱一致性以及最终一致性，具体介绍如下。

(1) 强一致性是指对数据完成更新操作之后，所有客户端访问到的数据均为更新之后的数据，这样可以保证客户端获取的是最新数据，但是若要达到强一致性，将会降低性能。

(2) 弱一致性是指当数据完成更新操作之后，系统并不保证所有的客户端访问到的数据都是最新数据，但是会尽量保证在某个时间(如秒级或分钟级)内让数据达到一致性状态。

(3) 最终一致性是弱一致性的一种特例，当对数据更新完之后，保证没有后续更新的前提下，系统最终返回的是上一次更新操作的值。

假设数据库中有一张名为 Personnel_Information 的数据表，该表中主要记录了 500 名职员的性别、身高、体重及健康值，具体如图 5-117 所示(注：这里只截取数据表中的部分数据进行展示)。

USERID	GENDER	HEIGHT	WEIGHT	INDEX
00000000001	Female	174	96	4
00000000002	Male	189	87	2
00000000003	Female	185	110	4
00000000004	Female	195	104	3
00000000005	Male	149	61	3
00000000006	Male	189	104	3
00000000007	Male	147	92	5
00000000008	Male	154	111	5
00000000009	Male	174	90	3
00000000010	Female	169	103	4
00000000011	Male	195	81	2
00000000012	Female	159	80	4
00000000013	Female	192	101	3
00000000014	Male	155	51	2
00000000015	Male	191	79	2
00000000016	Female	153	107	5
00000000017	Female	157	110	5
00000000018	Male	140	129	5
00000000019	Male	144	145	5
00000000020	Male	172	139	5

图 5-117 数据表 Personnel_Information

下面通过 Kettle 工具分步骤讲解使用弱一致性对数据表 Personnel_Information 中的数据进行一致性处理，即利用数据表 Personnel_Information 中的字段 GENDER 中的值训练出一个健康值预测模型，用于将原始数据中的字符串特征转换为模型可识别的数字特征，这里是将 GENDER 字段中的 Male 和 Female 转换成数字 0 和 1，然后将转换完的数据存储到新数据表 Personnel_ Information_New 中（注：新数据表 Personnel_ Information_ New 的数据结构应与原始数据表 Personnel_Information 的数据结构保持一致），具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 data _consistency，并添加“表输入”控件、“值映射”控件、“插入/更新”控件以及 Hop 跳连接线，具体效果如图 5-118 所示。

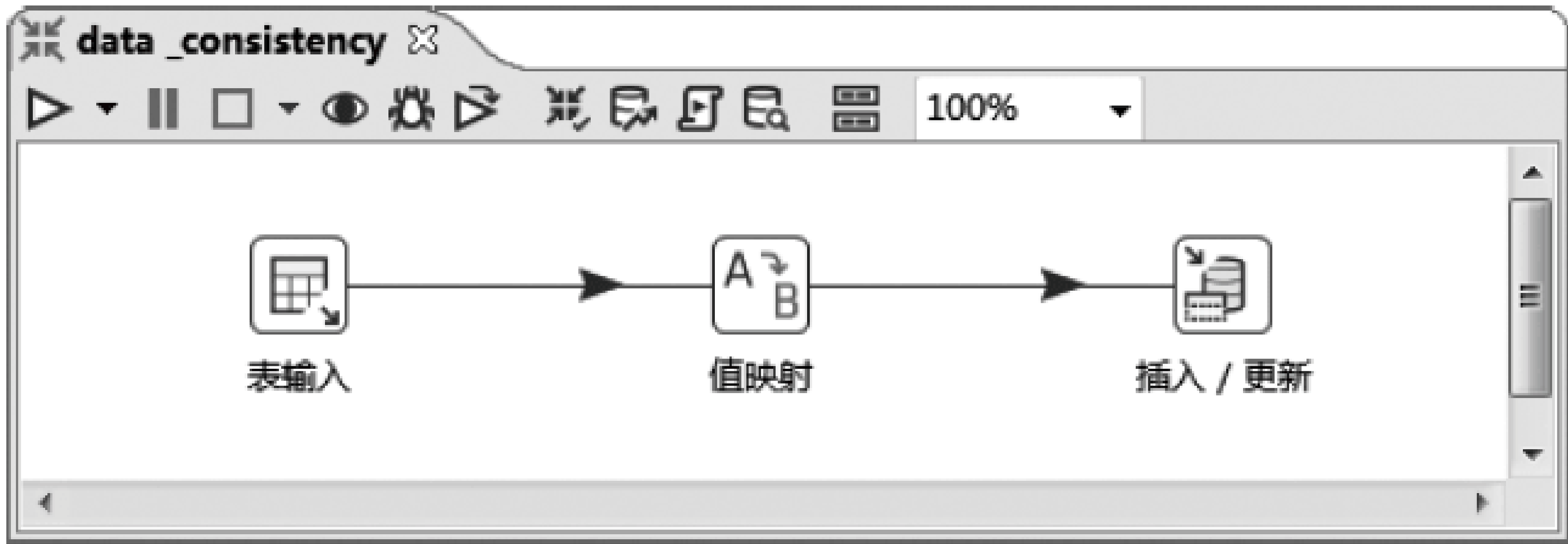


图 5-118　创建转换 data _consistency

2. 配置“表输入”控件

双击图 5-118 中的“表输入”控件，进入“表输入”界面，如图 5-119 所示。



图 5-119　“表输入”界面

在图 5-119 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 5-120 所示。

在图 5-119 中的 SQL 框中编写查询数据表 interpolation_data 的 SQL 语句，然后单击“预览”按钮，查看数据表 Personnel_Information 的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 5-121 和图 5-122 所示。



图 5-120　MySQL 数据库连接的配置



图 5-121　编写 SQL 语句

从图 5-122 中可以看出，数据表 Personnel_Information 的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“值映射”控件

双击图 5-118 中的“值映射”控件，进入“值映射”界面，如图 5-123 所示。

在图 5-123 中“使用的字段名”后的下拉列表中选择字段 GENDER；在“字段值”框中添加源值和目标值，这里是将 Male 替换成数字 0，将 Female 替换成数字 1，具体如图 5-124 所示。

在图 5-124 中单击“确定”按钮，完成“值映射”控件的配置。

预览数据

步骤 表输入 的数据 (501 rows)

#	USERID	GENDER	HEIGHT	WEIGHT	INDEX
1	1	Female	174	96	4
2	2	Male	189	87	2
3	3	Female	185	110	4
4	4	Female	195	104	3
5	5	Male	149	61	3
6	6	Male	189	104	3
7	7	Male	147	92	5
8	8	Male	154	111	5
9	9	Male	174	90	3
10	10	Female	169	103	4
11	11	Male	195	81	2
12	12	Female	159	80	4
13	13	Female	192	101	3
14	14	Male	155	51	2
15	15	Male	191	79	2
16	16	Female	153	107	5
17	17	Female	157	110	5
18	18	Male	140	129	5
19	19	Male	144	145	5
20	20	Male	172	139	5

关闭(C)　显示日志(L)

图 5-122　预览数据

值映射

步骤名称：值映射
使用的字段名：
目标字段名 (空=覆盖)
不匹配时的默认值:
字段值:

#	源值	目标值
1		

Help　确定(O)　取消(C)

图 5-123　“值映射”界面

值映射

步骤名称：值映射
使用的字段名：GENDER
目标字段名 (空=覆盖)
不匹配时的默认值:
字段值:

#	源值	目标值
1	Male	0
2	Female	1

Help　确定(O)　取消(C)

图 5-124　配置“值映射”控件

4. 配置“插入/更新”控件

双击图 5-118 中的“插入/更新”控件，进入“插入/更新”界面，如图 5-125 所示。

在图 5-125 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 5-126 所示。

单击图 5-125 中目标表右侧的“浏览”按钮，弹出“数据库浏览器”窗口，选择目标表 Personnel_ Information_New，具体如图 5-127 所示。

在图 5-127 中单击“获取字段”按钮，用来指定查询数据需要的关键字，这里选择的是 Personnel_ Information_New 数据表中的 USERID 字段和输入流里的 USERID 字段；单击

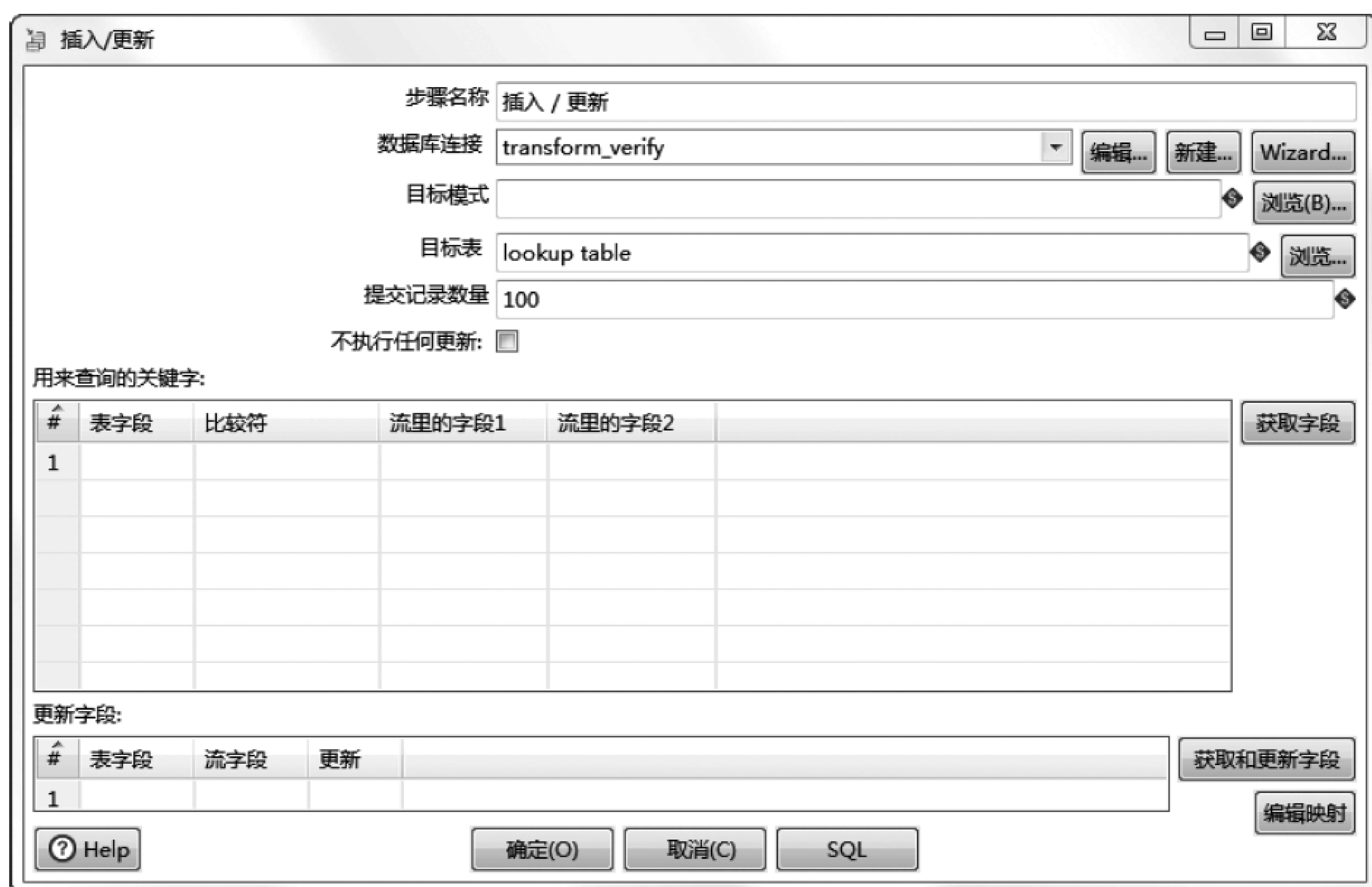


图 5-125 “插入/更新”界面



图 5-126 MySQL 数据库连接的配置

“获取和更新字段”按钮,用来指定需要更新的字段,具体如图 5-128 所示。

在图 5-128 中单击“确定”按钮,完成“插入/更新”控件的配置。

5. 运行转换 data _consistency

单击转换工作区顶部的 ▷ 按钮,运行创建的转换 data _consistency,实现将数据表

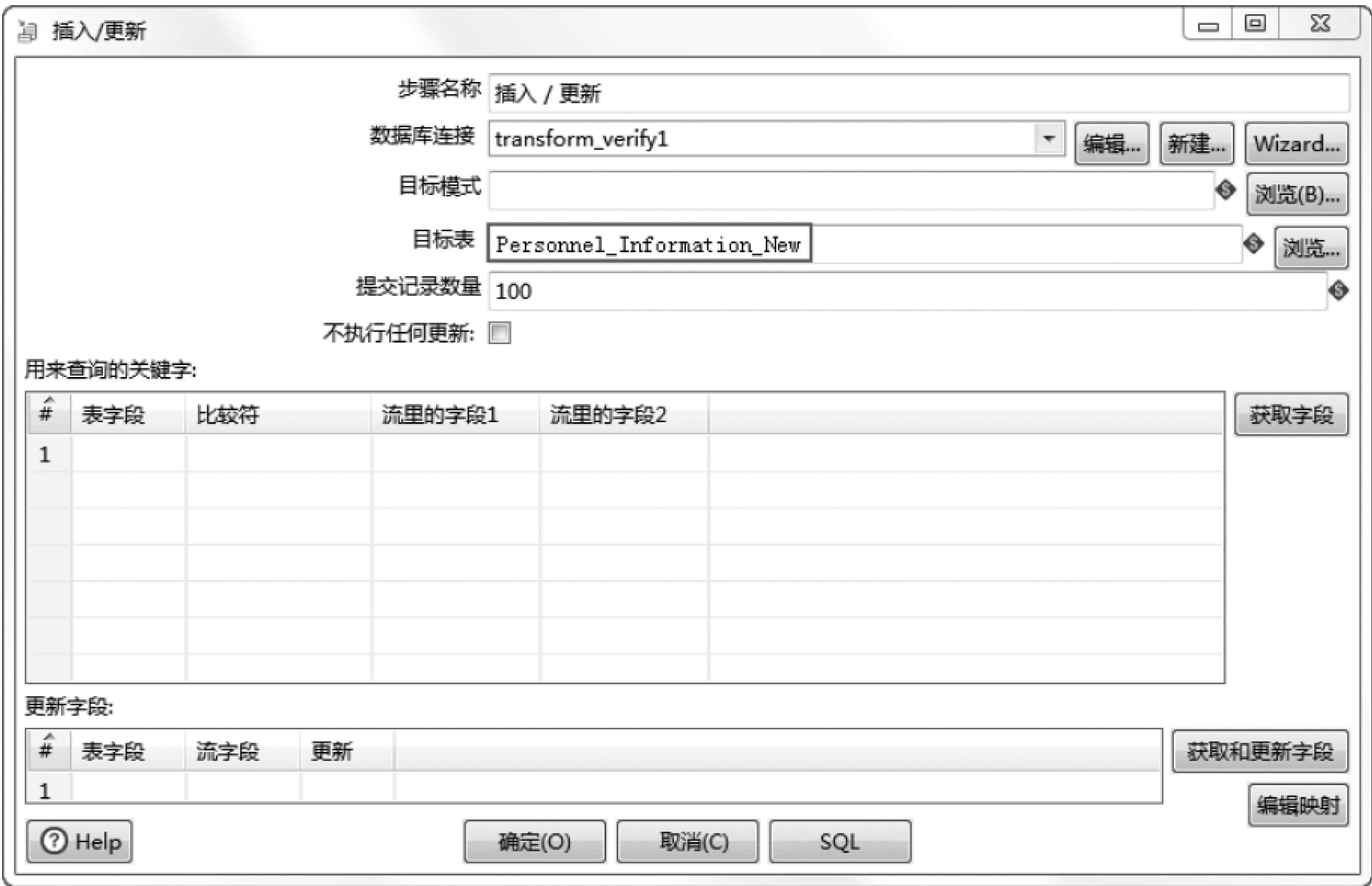


图 5-127 选择要插入数据的数据表 Personnel_ Information_New

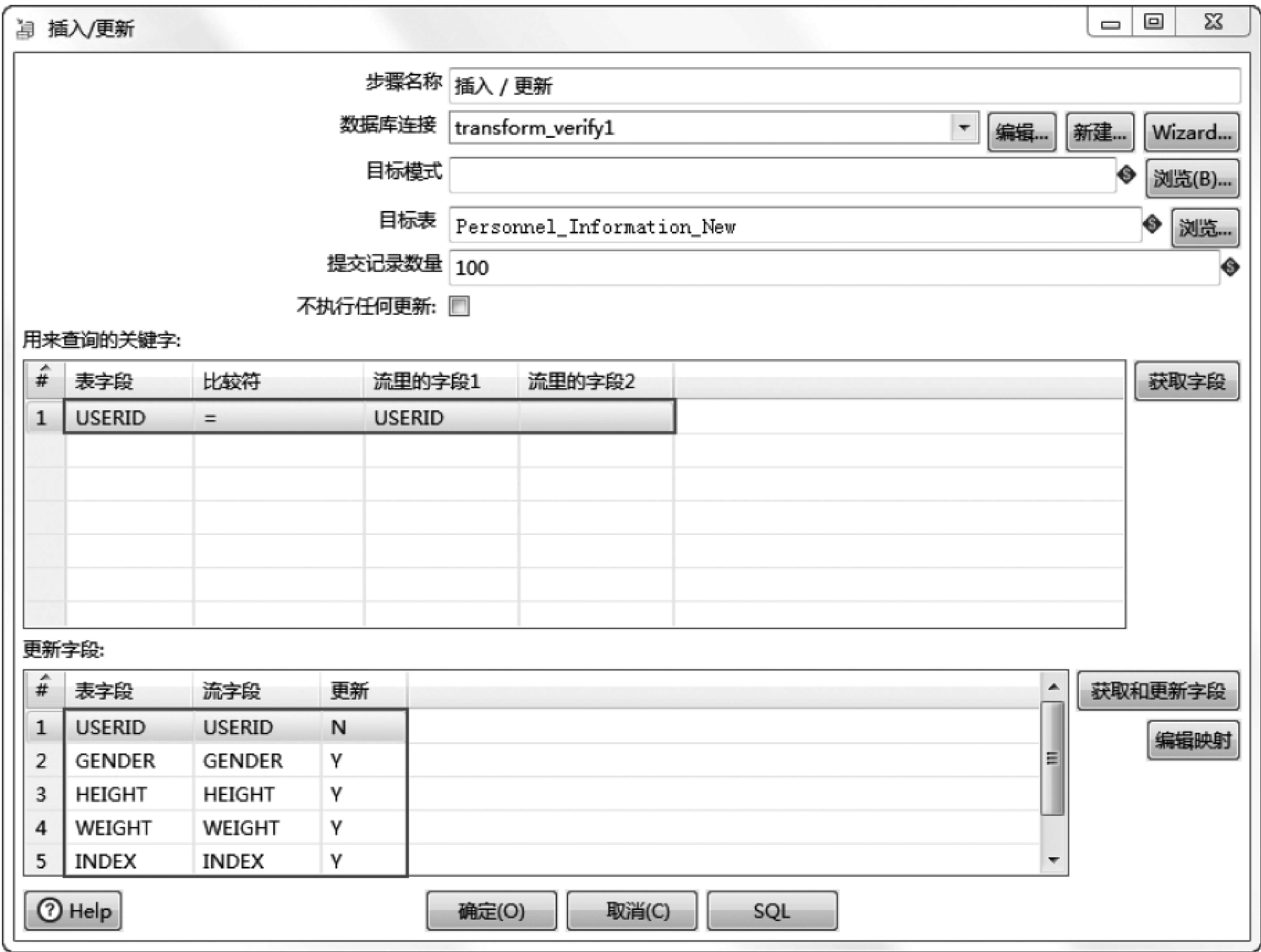
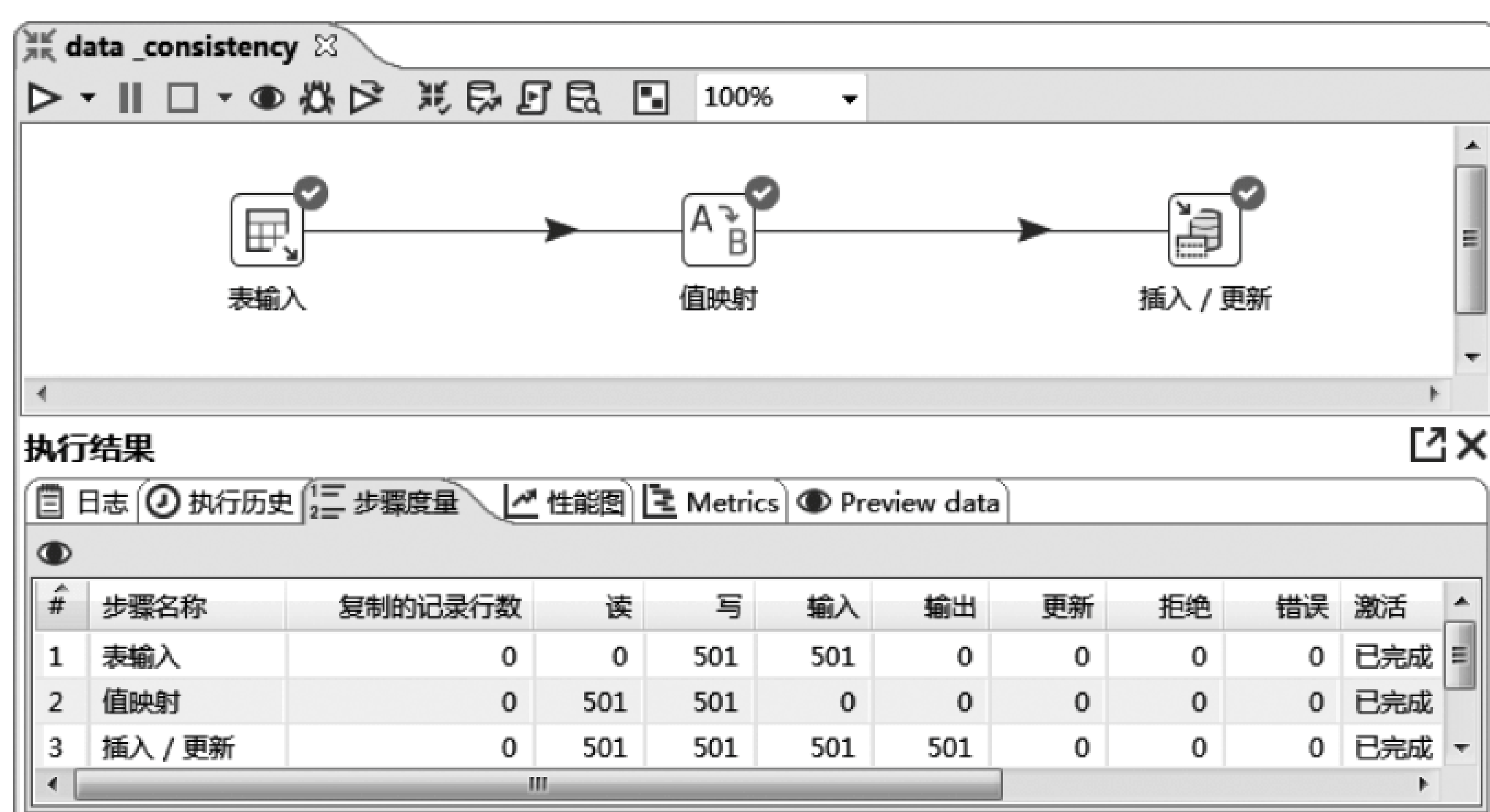


图 5-128 配置“插入/更新”控件

Personnel_Information 中的数据进行转换并更新到新的数据表 Personnel_ Information_New 中,具体如图 5-129 所示。

从图 5-129 中执行结果的“步骤度量”可以看出,“表输入”控件输入 501 条数据并写入



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	表输入	0	0	501	501	0	0	0	0	已完成
2	值映射	0	501	501	0	0	0	0	0	已完成
3	插入 / 更新	0	501	501	501	501	0	0	0	已完成

图 5-129　运行转换 data _consistency

该控件；“值映射”控件读取“表输入”控件中的 501 条数据并写入该控件；“插入/更新”控件读取“值映射”控件中的 501 条数据并写入该控件，将写入的数据与新数据表输入的 501 条数据进行比较，将 501 条数据的比较结果进行输出。

6．查看数据表 Personnel_ Information_New 中的数据

通过 SQLyog 工具，查看数据表 Personnel_ Information_New 是否已成功插入 501 条数据，查看结果如图 5-130 所示。

1 信息　2 表数据　3 信息

USERID	GENDER	HEIGHT	WEIGHT	INDEX
00000000001	1	174	96	4
00000000002	0	189	87	2
00000000003	1	185	110	4
00000000004	1	195	104	3
00000000005	0	149	61	3
00000000006	0	189	104	3
00000000007	0	147	92	5
00000000008	0	154	111	5
00000000009	0	174	90	3
00000000010	1	169	103	4
00000000011	0	195	81	2
00000000012	1	159	80	4
00000000013	1	192	101	3
00000000014	0	155	51	2
00000000015	0	191	79	2
00000000016	1	153	107	5
00000000017	1	157	110	5
00000000018	0	140	129	5
00000000019	0	144	145	5
00000000020	0	172	139	5

数据库： transform　表格： personnel_information_new

图 5-130　数据表 Personnel_ Information_New

从图 5-130 中可以看出，数据表 Personnel_ Information_New 中已插入 501 条数据（这

里只展示数据表中的部分数据)，说明我们成功实现了将数据表 Personnel_Information 中的数据进行转换操作，并将转换操作完的数据插入新数据表 Personnel_ Information_New 中。

由于转换 data _consistency 的运行只是单次的，若后续需要将原始数据进行转换、插入操作，则需要运行该转换，这样工作效率很低，因此，通过 Kettle 工具创建一个作业，对转换 data _consistency 设置定时器，使得转换 data _consistency 程序定时自动执行数据同步的操作。

7. 打开 Kettle 工具，创建作业

使用 Kettle 工具创建作业 data _consistency_job，并添加 Start 控件、“转换”控件以及作业跳连接线，具体效果如图 5-131 所示。

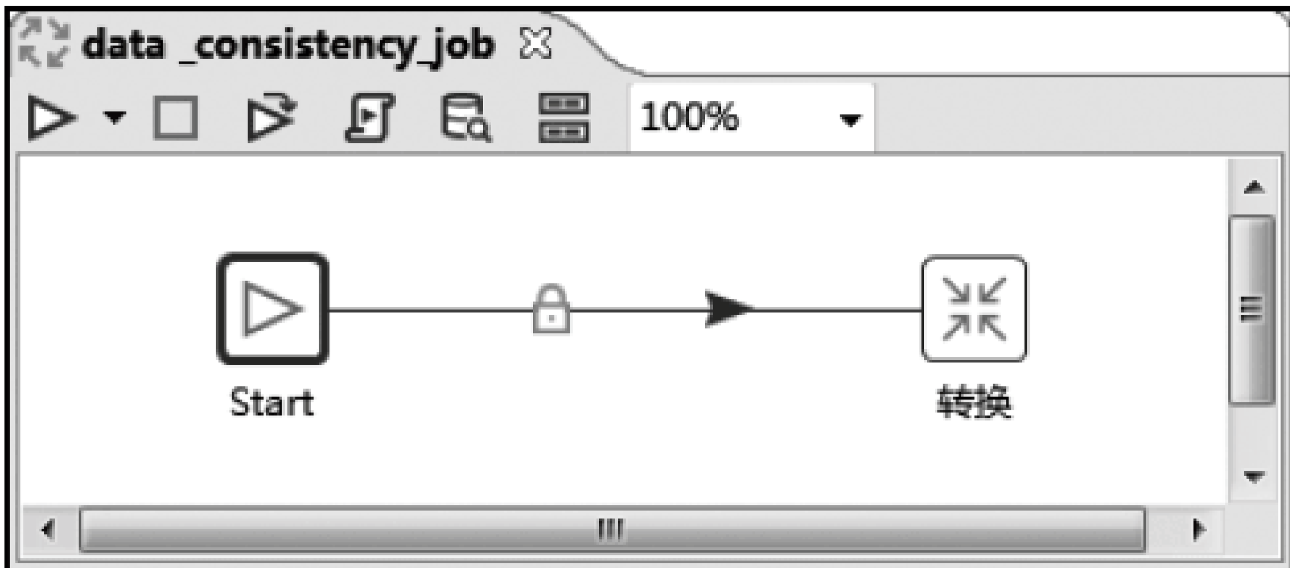


图 5-131　创建作业 data _consistency_job

8. 配置 Start 控件

双击图 5-131 中的 Start 控件，进入“作业定时调度”界面，具体如图 5-132 所示。

在图 5-132 中勾选“重复”复选框；在“类型”后的下拉列表中选择“时间间隔”定时，并设置“以秒计算的间隔”是 5，“以分钟计算的间隔”是 0(即每 5 秒运行一次转换，实现数据同步)，具体如图 5-133 所示。



图 5-132　“作业定时调度”界面



图 5-133　设置执行转换的时间间隔

在图 5-133 中单击“确定”按钮，完成 Start 控件的配置。

9. 配置“转换”控件

双击图 5-131 中的“转换”控件，进入“转换”界面，具体如图 5-134 所示。



图 5-134 “转换”界面

在图 5-134 中单击“浏览”按钮,选择添加转换 data _consistency 至作业中,如图 5-135 所示。



图 5-135 添加转换 data _consistency 至作业中

在图 5-135 中单击“确定”按钮,完成“转换”控件的配置。

10. 运行作业 data _consistency_job

单击作业工作区顶部的▷按钮,运行创建的作业 data _consistency_job,实现让数据表 Personnel_Information 中的数据定时进行转换,并更新到数据表 Personnel_ Information_ New 中,具体如图 5-136 所示。

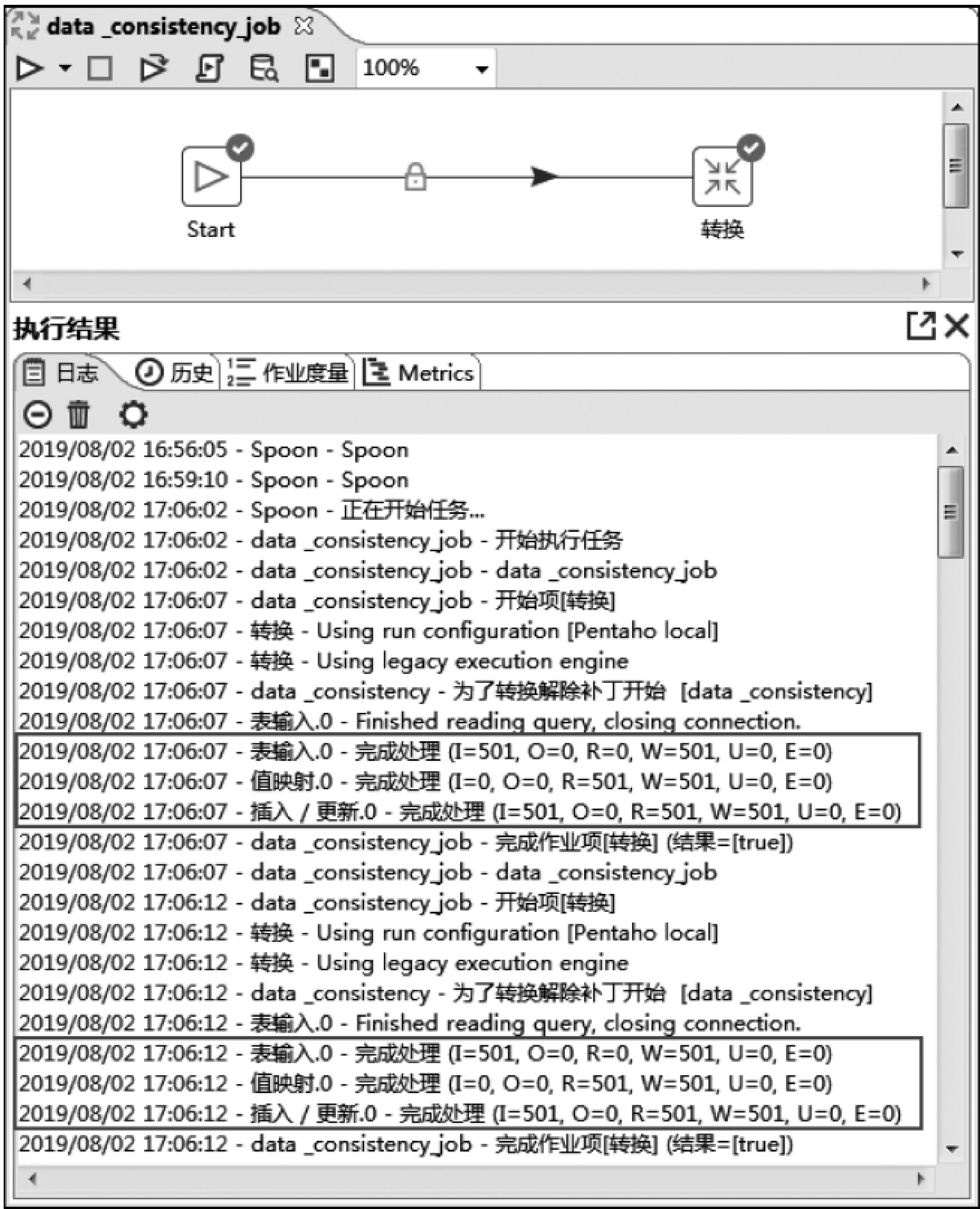


图 5-136　运行作业 data _consistency_job

从图 5-136 中控制台的日志输出可以看出,作业每隔 5 秒执行一次转换,实现数据同步并且在执行过程中会记录处理的数据量以及当前时间段处理的数据条数,说明已经成功实现了让数据表 Personnel_Information 中的数据定时进行转换,并更新到数据表 Personnel_Information_New 中。

5.4.2　数据规范化处理

由于数据源系统分散在各个业务线上,不同业务线对数据的要求、理解和规范也不同,这样就会导致对同一数据对象的描述规格完全不同,因此,在数据清洗的过程中需要将统一数据规范的数据抽取出来进行规范处理。

为了提高数据的可读性及合理性,企业会要求数据遵守一定的规范,具体规范如下。

(1) 电子邮箱的地址必须是有效的格式。

(2) 输入的数据都必须是大写/小写。

(3) 日期必须是 dd-mm-yyyy 的格式。

(4) 电话号码必须是 xxx-xxxx-xxxx 的格式。

(5) 用户的年龄必须大于 18 岁。

(6) 数值不能超过预定义的值。

综上所述,这些规范都有一个共同点,即检查数据都必须遵守预定义的业务规则,找出不符合业务规则的数据。

下面通过 Kettle 工具分步骤讲解如何对数据进行检验操作,具体步骤如下。

1. 打开 Kettle 工具,创建转换

使用 Kettle 工具创建转换 data_validation,并添加"自定义常量数据"控件、"计算器"控件、"数据检验"控件、"空操作"控件以及 Hop 跳连接线,具体效果如图 5-137 所示。

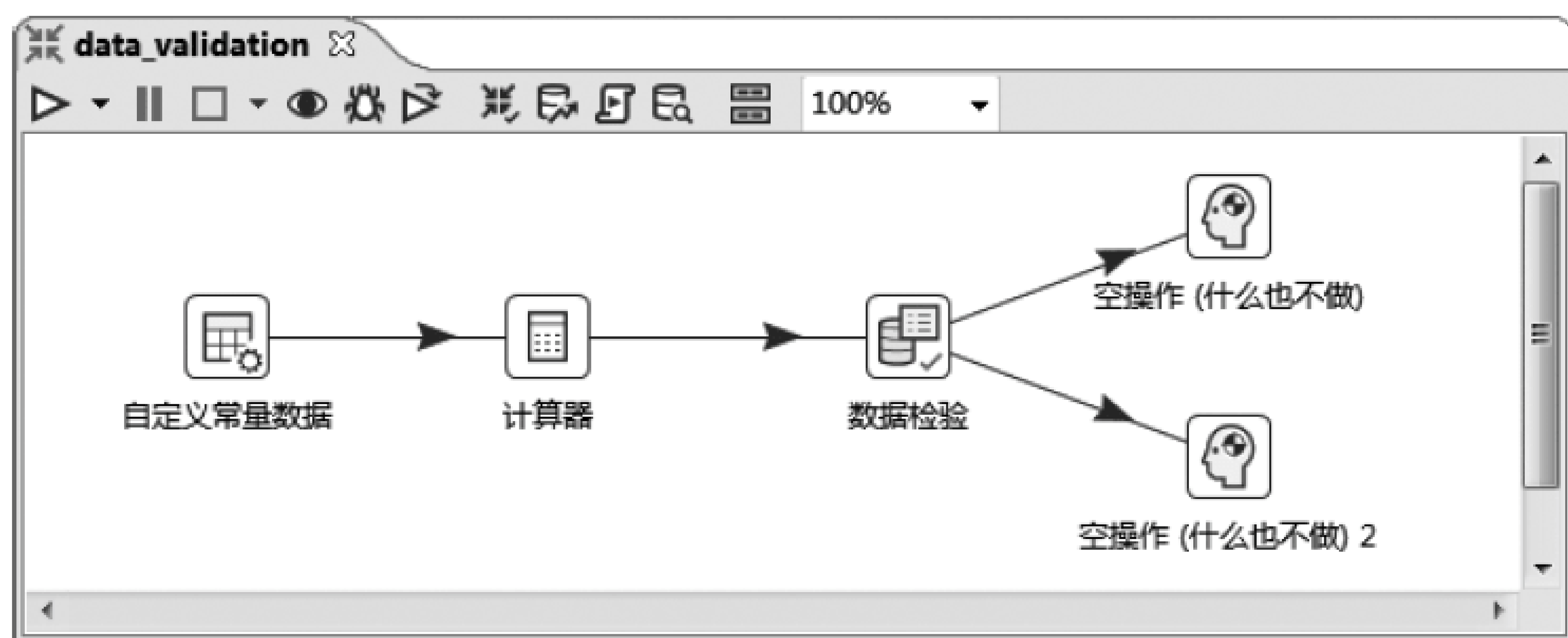


图 5-137 创建转换 data_validation

2. 配置"自定义常量数据"控件

双击图 5-137 中的"自定义常量数据"控件,进入"自定义常量数据"界面配置实验用数据,如图 5-138 所示。



图 5-138 "自定义常量数据"界面

在图 5-138 中单击"元数据"选项卡,添加字段常量 ProductionDate、ProductionName、ProductionNumber 以及 CommoditySales 并指定其数据类型;单击"数据"选项卡,添加自定义的数据,具体效果如图 5-139 所示。

在图 5-139 中单击"确定"按钮,完成"自定义常量数据"控件的配置。



图 5-139　“自定义常量数据”控件配置的效果图

3. 配置“计算器”控件

双击图 5-137 中的“计算器”控件，进入“计算器”界面，如图 5-140 所示。



图 5-140　“计算器”界面

在图 5-140 中的“字段”处添加一个新字段 UnitPrice，用于存储计算出的产品单价数据；在“字段 A”和“字段 B”处的下拉选项中分别选择 CommoditySales（销售额）和 ProductionNumber（销售数量）字段；在“计算”处的下拉框中选择“A/B”，即表示将字段 A 与字段 B 进行相除计算，具体如图 5-141 所示。

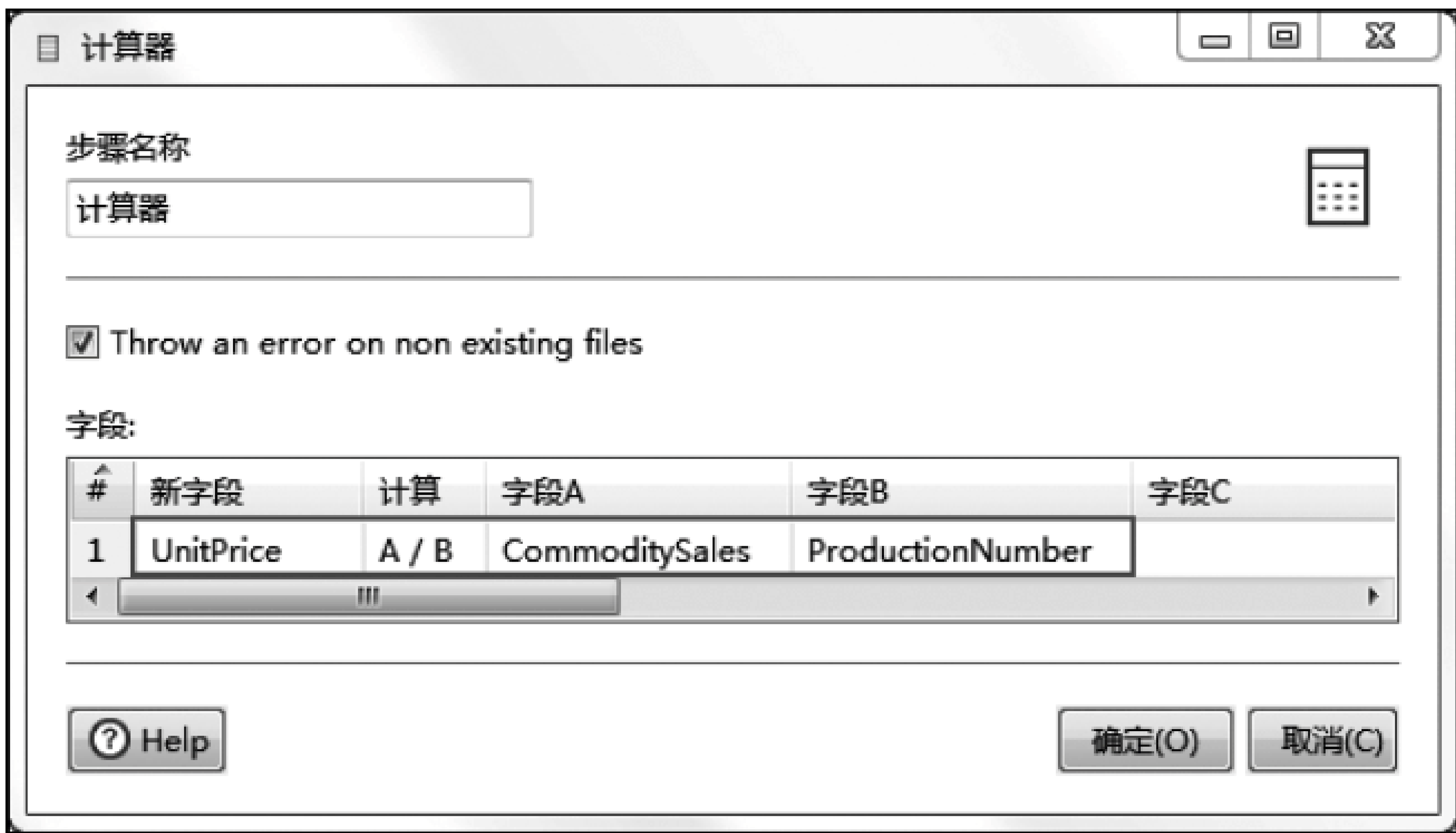


图 5-141　“计算器”控件配置的效果图

在图 5-141 中单击“确定”按钮，完成“计算器”控件的配置。

4. 配置“数据检验”控件

双击图 5-137 中的“数据检验”控件，进入“数据检验”界面，如图 5-142 所示。



图 5-142 “数据检验”界面

在图 5-142 中单击“增加检验”按钮，增加检验条件，这里我们制定的检验条件有 3 个，即日期(ProductionDate)不能在 2019 年 1 月 1 日之前、产品名称(ProductionNmae)必须是小写以及单个产品价格(UnitPrice)不能超过 10。

单击图 5-142 中的“增加检验”按钮，弹出“数据检验”窗口，在该窗口中添加检验名称 date_verify 用于校验日期，添加后单击“确定”按钮关闭“数据检验”窗口，效果如图 5-143 所示。

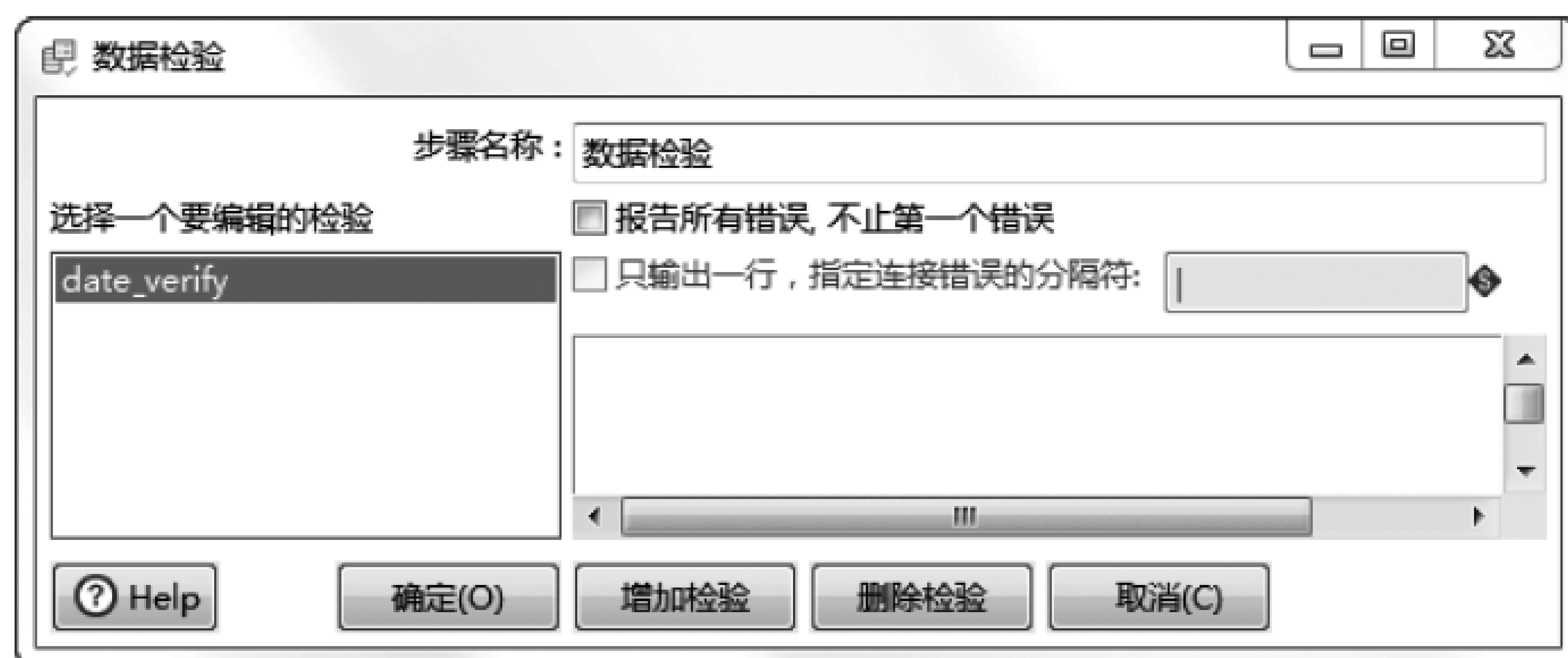


图 5-143 指定第一个检验条件的名称为 date_verify

在图 5-143 中单击检验 date_verify，右边的空白框中会出现配置第一个检验条件的相关参数，具体如图 5-144 所示。

在图 5-144 中的“要检验的字段名”处添加要检验的字段；在“错误代码”和“错误描述”处自定义检验到错误数据时日志的输出内容；勾选“检验数据类型”复选框；在“数据类型”处指定数据类型，用于判断指定检验字段的数据类型与这里指定的数据类型是否一致；在“转换掩码”处输入与指定检验字段相同的日期格式，若指定的检验字段日期格式与 Kettle 默认的日期格式不同，则报错；在“最小值”处添加检验条件，具体如图 5-145 所示。

单击图 5-145 中的“增加检验”按钮，弹出“数据检验”窗口，在该窗口中添加检验名称 name_verify 用于检验商品名称，添加后单击“确定”按钮关闭“数据检验”窗口，效果如图 5-146 所示。

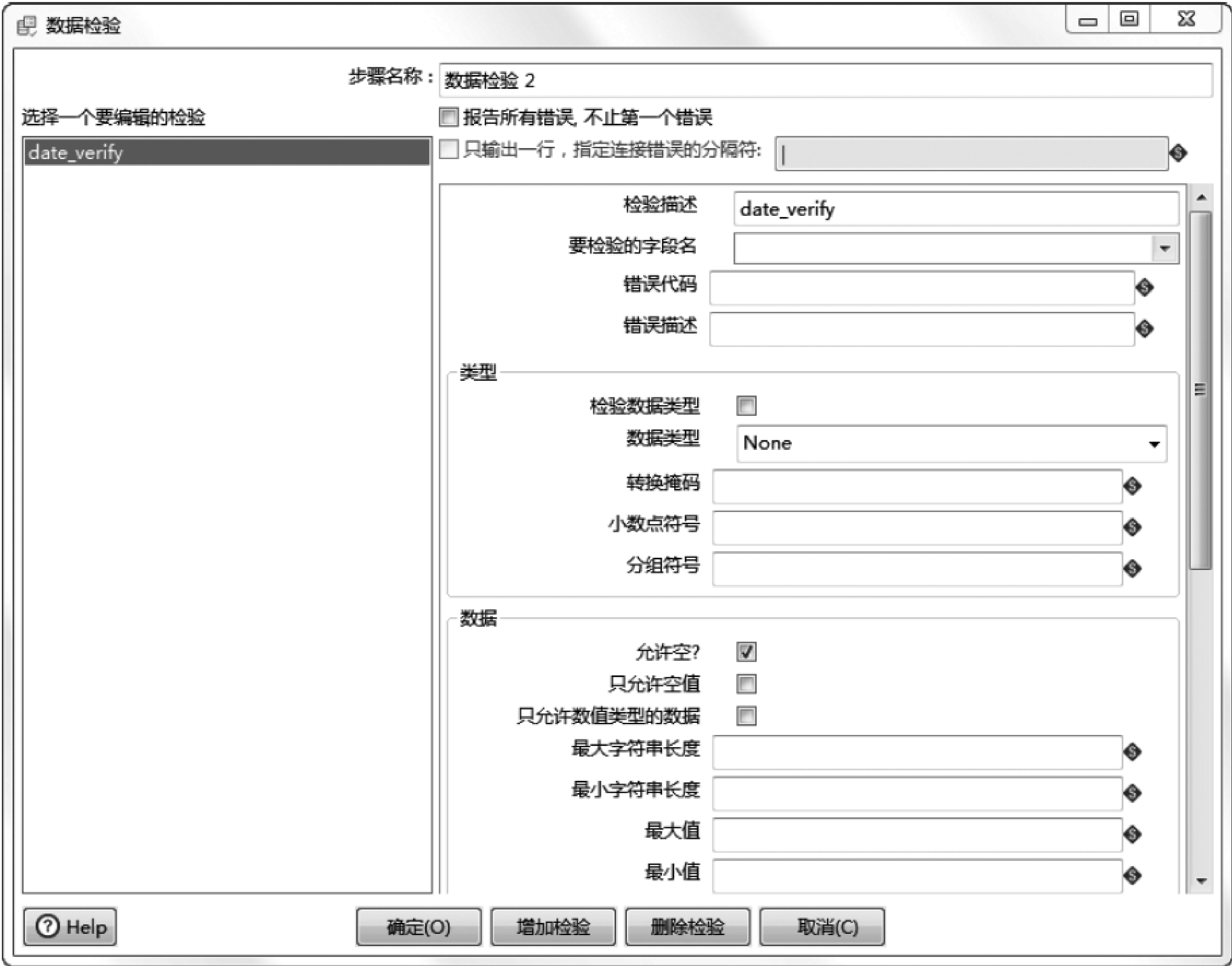


图 5-144　配置第一个检验条件的相关参数



图 5-145　第一个检验条件的配置

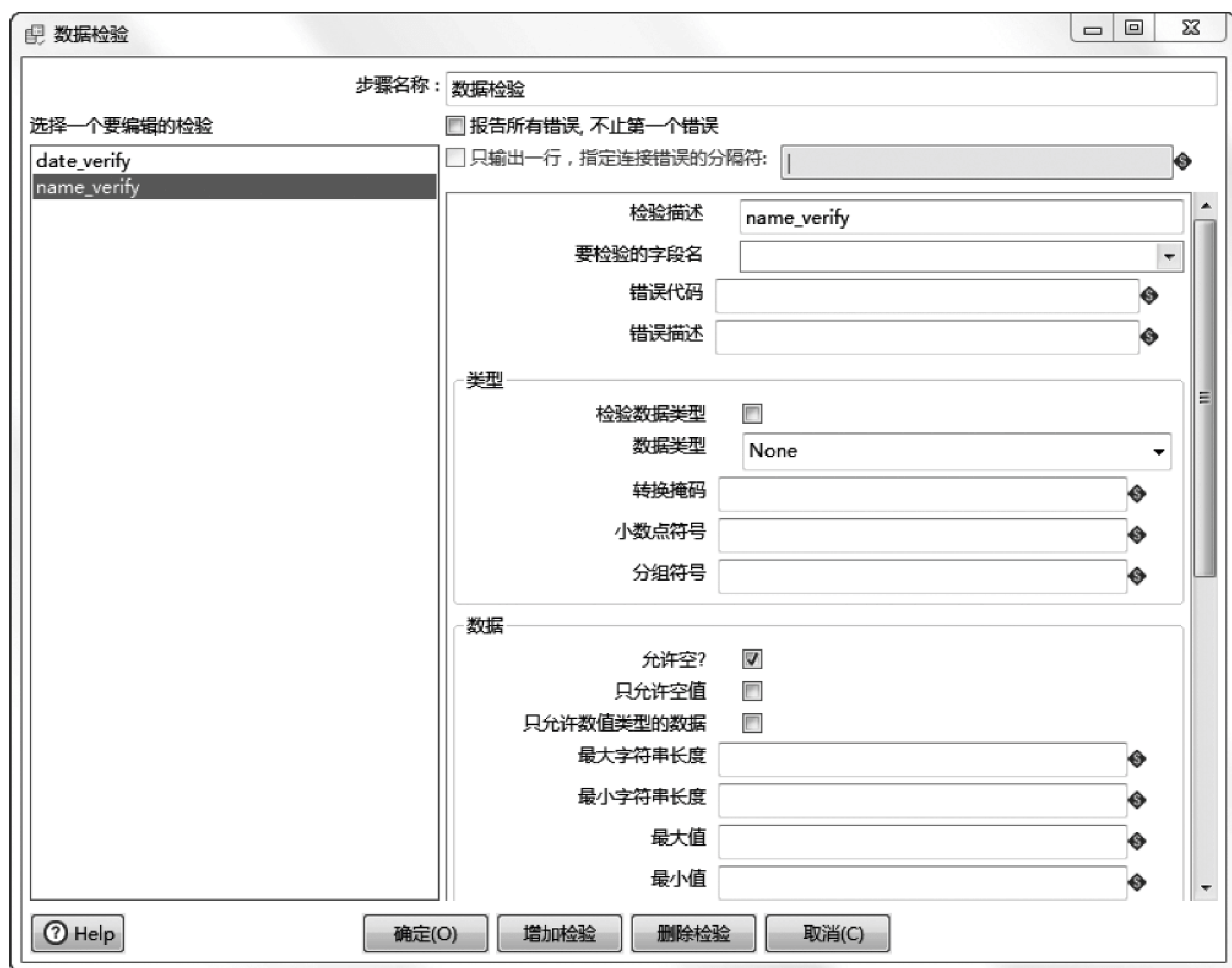


图 5-146　指定第二个检验条件的名称为 name_verify

在图 5-146 中的“要检验的字段名”处添加要检验的字段；在“错误代码”和“错误描述”处自定义检验到错误数据时日志的输出内容；在“合法数据的正则表达式”处添加检验条件进行判断，这里添加的正则表达式用于检验产品名称是否为小写，具体如图 5-147 所示。

单击图 5-147 中的“增加检验”按钮，弹出“数据检验”窗口，在该窗口中添加检验名称 price_verify 用于检验产品单价，添加后单击“确定”按钮关闭“数据检验”窗口，效果如图 5-148 所示。

在图 5-148 中的“要检验的字段名”处添加要检验的字段；在“错误代码”和“错误描述”处自定义检验到错误数据时日志的输出内容；勾选“检验数据类型”复选框；在“数据类型”处指定数据类型；在“小数点符号”处添加小数点符号，即“.”；在“最大值”处添加检验条件，具体如图 5-149 所示。

在图 5-149 中单击“确定”按钮，完成“数据检验”控件的配置。

5．配置“空操作”控件

单击选中图 5-137 中的“数据检验”控件，然后按住 Shift 键，通过分发方式设置“主输出步骤”连接到“空操作(什么也不做)”控件；设置“错误处理步骤”连接到“空操作(什么也不做)2”控件，如图 5-150 所示。



图 5-147　配置第二个检验条件的参数



图 5-148　指定第三个检验条件的名称为 price_verify

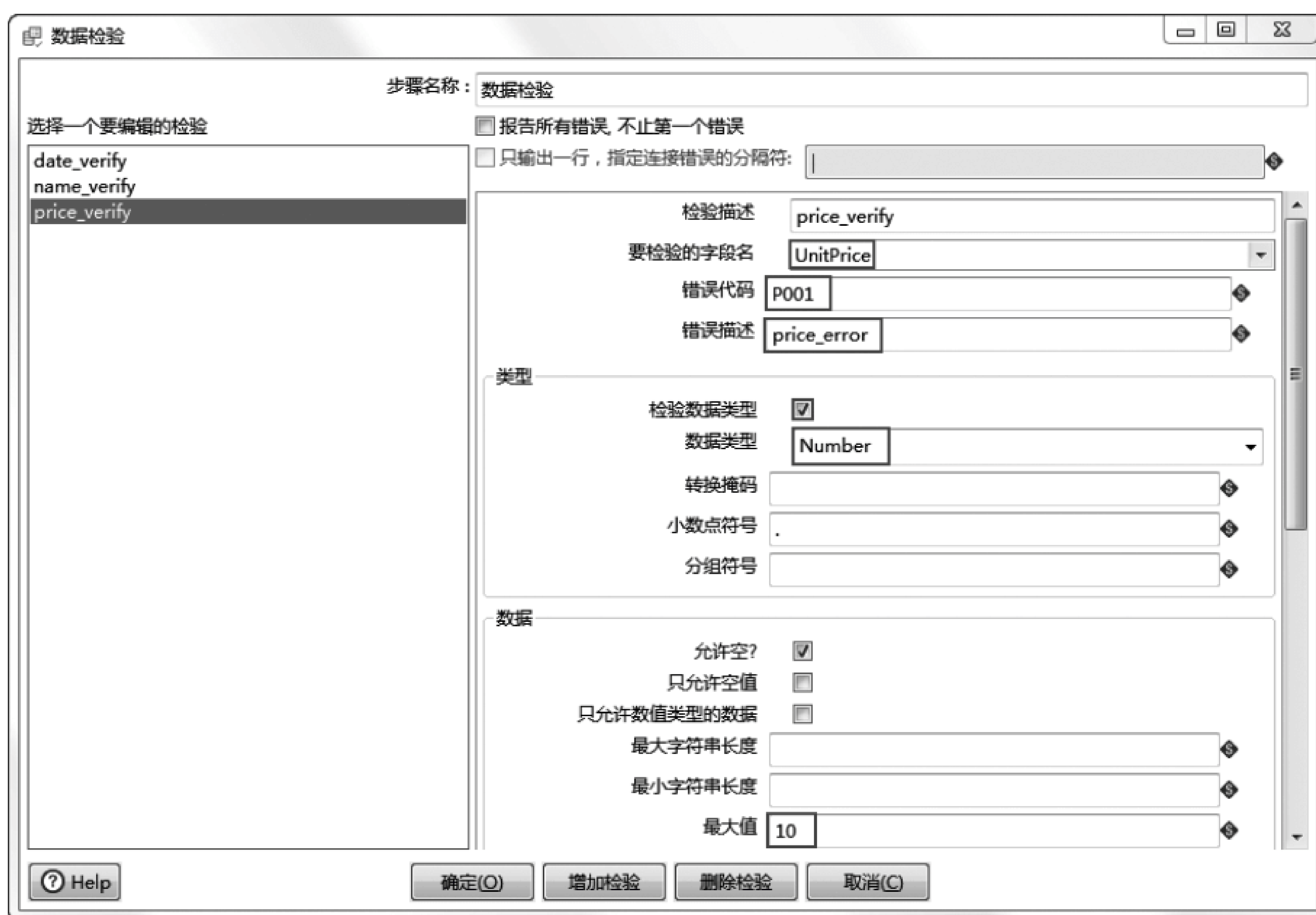


图 5-149　配置第三个检验条件的参数

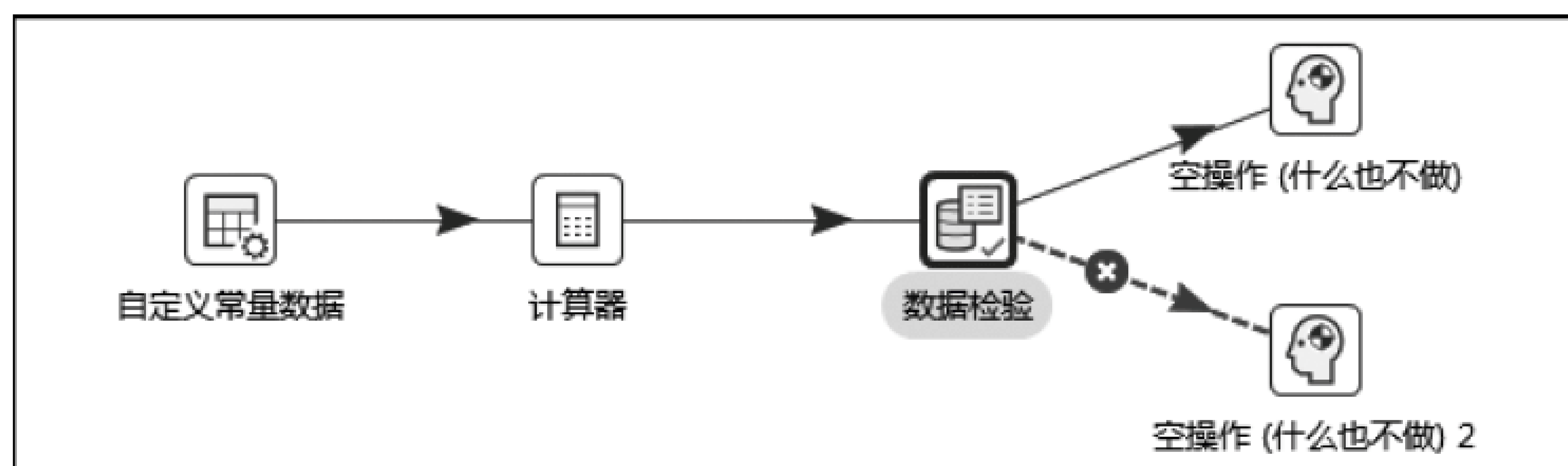


图 5-150　配置“空操作”控件

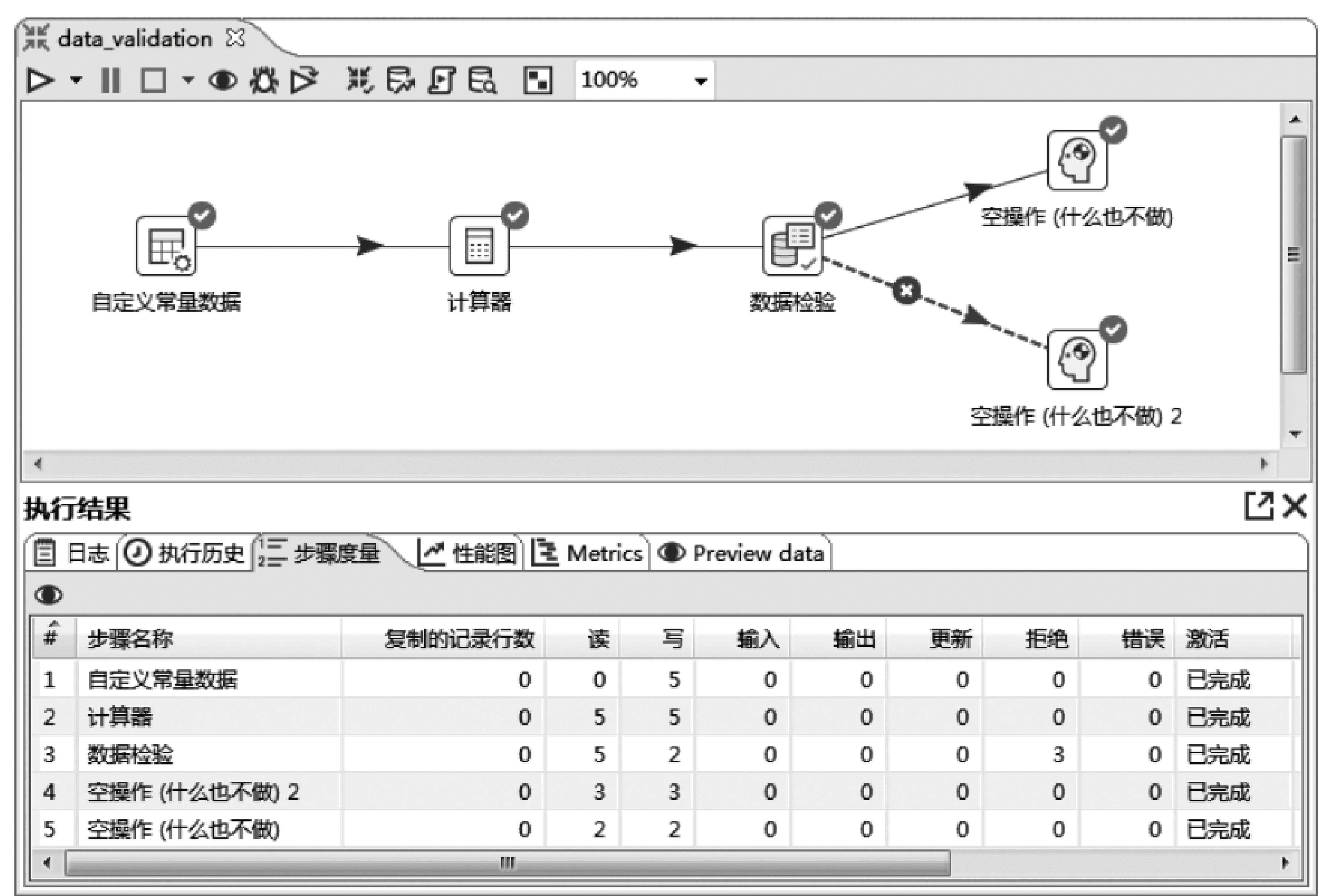
6. 运行转换 data_validation

单击转换工作区顶部的▷按钮，运行创建的转换 data_validation，实现对数据进行检验操作，具体如图 5-151 所示。

从图 5-151 中执行结果的“步骤度量”可以看出，“自定义常量数据”控件写入 5 条数据；“计算器”控件读取“自定义常量数据”控件的 5 条数据并写入该控件；“数据检验”控件读取“计算器”控件的 5 条数据，将符合校验要求的 2 条数据写入该控件；“空操作(什么也不做)2”控件读取并写入不符合校验要求的 3 条数据；“空操作(什么也不做)”控件读取并写入符合要求的 2 条数据。

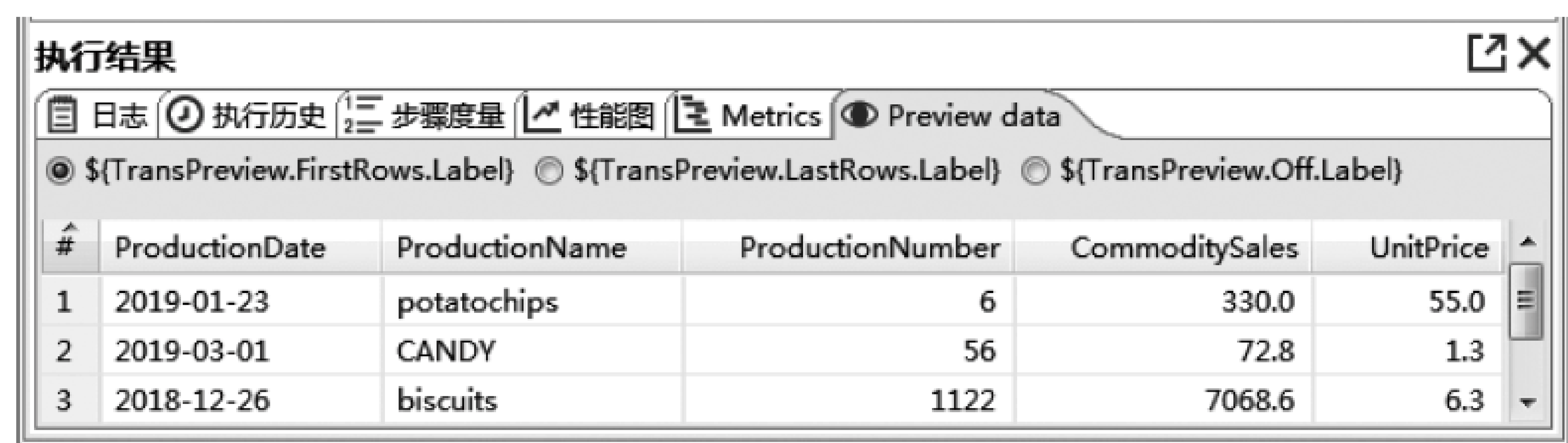
选中图 5-151 中的“空操作(什么也不做)2”控件，单击执行结果窗口的 Preview data 选项卡，查看是否将不符合校验规则的数据检验出来，具体如图 5-152 所示。

从图 5-152 中可以看出，原始数据中不符合校验规则的 3 条数据均被检验出来，说明我们通过 Kettle 工具实现了数据检验的功能。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	自定义常量数据	0	0	5	0	0	0	0	0	已完成
2	计算器	0	5	5	0	0	0	0	0	已完成
3	数据检验	0	5	2	0	0	0	3	0	已完成
4	空操作 (什么也不做) 2	0	3	3	0	0	0	0	0	已完成
5	空操作 (什么也不做)	0	2	2	0	0	0	0	0	已完成

图 5-151　运行转换 data_validation



#	ProductionDate	ProductionName	ProductionNumber	CommoditySales	UnitPrice
1	2019-01-23	potatochips	6	330.0	55.0
2	2019-03-01	CANDY	56	72.8	1.3
3	2018-12-26	biscuits	1122	7068.6	6.3

图 5-152　查看是否将不符合校验规则的数据检验出来

5.5　本章小结

本章主要讲解数据清洗与检验的相关知识，包括数据去重、缺失值处理、异常值处理以及数据检验。希望读者通过本章的学习，可以掌握对重复数据、缺失值数据、异常值数据的处理，也可以掌握对数据进行检验操作。

5.6　本章习题

一、填空题

1. 常见的数据清洗操作包括重复值的处理、____________、异常值的处理。
2. 数据缺失分为两种，分别是行记录的缺失和____________。
3. 异常值的检测方法通常分为三大类，即________、监督式异常值的检测以及________。

4. 数据的一致性有 3 种类型，即强一致性、________以及最终一致性。

5. 修补异常值的方式主要有两种，即________和替换异常值。

二、判断题

1. 完全去重指的是消除不完全重复的数据。（　　）

2. 缺失值产生的原因主要是人为原因。（　　）

3. 箱形图又称为箱线图，是一种用于显示一组数据分散情况的统计图。（　　）

4. 数据一致性是指在对一个副本数据进行更新的同时，无须确保也能更新到其他副本。（　　）

5. 检查数据都必须遵守预定义的业务规则，找出不符合业务规则的数据。（　　）

三、选择题

1. 下列方法中，________不是填充缺失值的方法。

A. 均值填充　　B. 热卡填充　　C. 3σ 准则　　D. 回归填充

2. 下列规范中，为了提高数据的可读性及合理性，企业会要求数据遵守________。

A. 电子邮箱的地址必须是有效的格式

B. 用户的年龄必须小于 18 岁

C. 数值可超过预定义的值

D. 电话号码无须是 xxx-xxxx-xxxx 的格式

3. 下列策略中，________不属于修改异常值的策略。

A. 最邻近值替代异常值　　B. 均值替代异常值

C. 众数替代异常值　　D. 异常值替换成缺失值

四、操作题

通过 Kettle 工具，实现以下功能：

(1) 对文件 merge.csv 进行完全去重。

(2) 对文件 people_survey.txt 中的缺失值进行填充。

第6章 数据转换

学习目标

（1）熟悉多数据源的合并

（2）掌握不一致数据转换

（3）掌握数据粒度的转换

（4）掌握数据的合计处理

数据的清洗过程除了包括第5章提到的对数据本身的清洗与检验操作，还包括数据转换操作。数据转换是数据清洗过程的重要步骤之一，它的主要任务是进行不一致数据的转换、数据粒度的转换，以及一些商务规则的计算。本章将针对数据转换的相关知识进行详细讲解。

6.1 多数据源的合并

随着信息技术的发展和科技的进步，人类步入大数据时代，大数据作为当前高科技时代的产物，它的种类多而繁杂。如果想得到需要的数据，这些需要的数据有可能来源于多个不同的数据源中，此时我们可以将多个数据源进行合并操作，从而获取到所需要的数据。

假设某公司旗下有两个子公司，分别为A公司和B公司，且这两个子公司均在销售手机，其中A公司的手机日销售情况存储在CSV文件中，即文件company_a.csv；B公司的手机日销售情况存储在数据库的数据表中，即数据表company_b，具体内容如图6-1和图6-2所示。

下面通过Kettle工具将A公司和B公司的手机日销售数据合并到一个数据源(数据表company)中，也就是对文件company_a.csv和数据表company_b中的数据进行合并操作，并输出到数据表company中，具体实现步骤如下。

1．打开Kettle工具，创建转换

使用Kettle工具创建转换company_merge，并添加“CSV文件输入”控件、“表输入”控件、“字段选择”控件、“排序合并”控件、“增加序列”控件、“表输出”控件以及Hop跳连接线，具体效果如图6-3所示。

2．配置“CSV文件输入”控件

双击图6-3中的“CSV文件输入”控件，进入“CSV文件输入”界面，具体如图6-4所示。

id	salesArea	brand	model	unitPrice	number
1	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
2	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
3	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
4	石家庄市	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
5	大连市	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
6	秦皇岛市	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
7	青岛市	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
8	西安市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
9	大同市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
10	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9699	38
11	哈尔滨市	OPPO	OPPO R17（8GB RAM/全网通）	2499	29
12	深圳市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	36
13	西安市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	15
14	太原市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	24
15	天津市	三星	三星GALAXY Note 10（8GB/256GB/全网通）	6599	23
16	武汉市	小米	小米9（8GB/256GB/全网通）	2999	37
17	秦皇岛市	vivo	vivo X27 Pro（8GB RAM/全网通）	3598	33
18	大连市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	18
19	上海市	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	42
20	北京市	华为	华为Mate30 Pro（8GB/256GB/全网通/5G版/玻璃版）	6899	35

图 6-1 文件 company_a. csv 的数据内容

id	salesArea	brand	model	unitPrice	number
1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
3	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
4	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
5	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
6	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
7	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
8	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
9	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
10	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39
11	上海市	华为	华为Mate 20 X（8GB/256GB/全网通/5G版）	6199	42
12	北京市	三星	三星Galaxy S10+（8GB RAM/陶瓷版/全网通）	7499	27
13	河北省	苹果	苹果iPhone 11（4GB/128GB/全网通）	5999	20
14	浙江省	苹果	苹果iPhone 11（4GB/256GB/全网通）	6799	29
15	天津市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	35
16	上海市	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	30
17	重庆市	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	38
18	山东省	华为	华为Mate30 Pro（8GB/128GB/全网通）	5799	20
19	陕西省	小米	小米MIX 3（6GB RAM/全网通）	2299	33
20	山西省	华为	华为Mate30 Pro（8GB/256GB/全网通/5G版/玻璃版）	6899	20

图 6-2 数据表 company_b 的数据内容

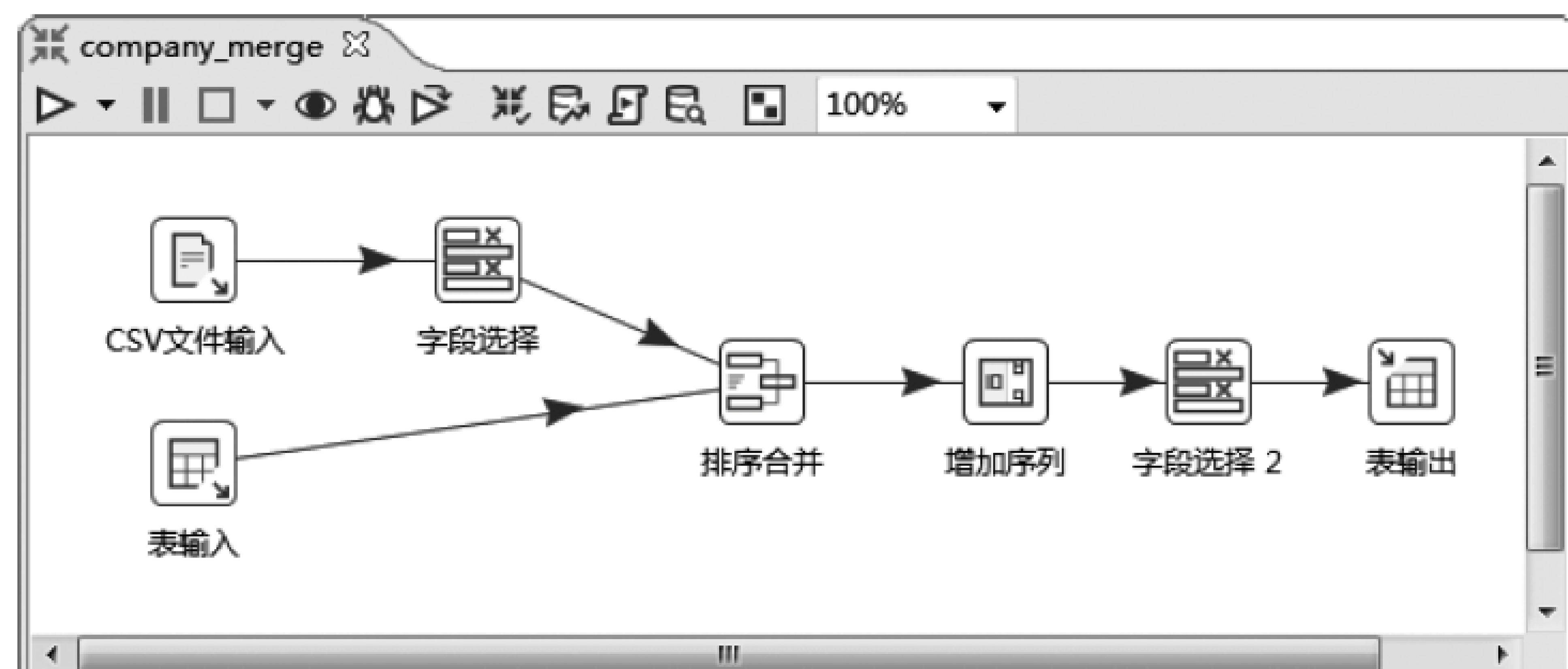


图 6-3 创建转换 company_merge



图 6-4　“CSV 文件输入”界面

单击图 6-4 中的“浏览”按钮，选择要抽取的 CSV 文件 company_a. csv；单击“获取字段”按钮，Kettle 自动检索 CSV 文件获取字段名称，并对文件中字段的类型、格式、长度、精度等属性进行解析，具体效果如图 6-5 所示。



图 6-5　配置“CSV 文件输入”控件

单击图 6-5 中的“预览”按钮，查看文件 company_a. csv 的数据是否抽取到 CSV 文件输入流中，具体效果如图 6-6 所示。

预览数据

步骤 CSV文件输入 的数据 (20 rows)

#	id	salesArea	brand	model	unitPrice	number
1	1	北京市	苹果	苹果iPhone 11 Pro (6GB/64GB/全网通)	8699	50
2	2	上海市	苹果	苹果iPhone 11 Pro (6GB/256GB/全网通)	9999	20
3	3	北京市	三星	三星GALAXY Note 10+ (12GB/256GB/全网通/5G版)	7999	32
4	4	石家庄市	OPPO	OPPO Reno2 (8GB/128GB/全网通)	2999	20
5	5	大连市	OPPO	OPPO Reno 10倍变焦版 (8GB/256GB/全网通)	4299	35
6	6	秦皇岛市	vivo	vivo NEX 3 (8GB/256GB/全网通/5G版)	5698	26
7	7	青岛市	vivo	vivo iQOO Pro (8GB/128GB/5G全网通)	3798	25
8	8	西安市	华为	华为P30 Pro (8GB/128GB/全网通)	4988	30
9	9	大同市	小米	小米9 Pro (8GB/256GB/全网通/5G版)	3799	22
10	10	重庆市	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9699	38
11	11	哈尔滨市	OPPO	OPPO R17 (8GB RAM/全网通)	2499	29
12	12	深圳市	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9599	36
13	13	西安市	苹果	苹果iPhone 11 Pro (6GB/256GB/全网通)	9999	15
14	14	太原市	华为	华为P30 Pro (8GB/128GB/全网通)	4988	24
15	15	天津市	三星	三星GALAXY Note 10 (8GB/256GB/全网通)	6599	23
16	16	武汉市	小米	小米9 (8GB/256GB/全网通)	2999	37
17	17	秦皇岛市	vivo	vivo X27 Pro (8GB RAM/全网通)	3598	33
18	18	大连市	苹果	苹果iPhone 11 Pro (6GB/64GB/全网通)	8699	18
19	19	上海市	华为	华为Mate30 (8GB/128GB/全网通/5G版/玻璃版)	4999	42
20	20	北京市	华为	华为Mate30 Pro (8GB/256GB/全网通/5G版/玻璃版)	6899	35

关闭(C) 显示日志(L)

图 6-6 预览数据

从图 6-6 中可以看出，CSV 文件 company_a. csv 的数据已经成功抽取到 CSV 文件输入流中，单击“关闭”→“确定”按钮，完成“CSV 文件输入”控件的配置。

3. 配置“字段选择”控件

双击图 6-3 中的“字段选择”控件，进入“选择/改名值”界面，如图 6-7 所示。

选择/改名值

步骤名称 字段选择

选择和修改 移除 元数据

字段：

#	字段名称	改名成	长度	精度
1				

获取选择的字段 列映射

包含未指定的列,按名称排序

Help 确定(O) 取消(C)

图 6-7 “选择/改名值”界面

在图 6-7 中"选择和修改"选项卡的"字段"处可以手动添加"CSV 文件输入"控件输出的所有数据字段，也可以单击"获取选择的字段"按钮，Kettle 工具自动检索并添加"CSV 文件输入"控件输出的所有数据字段，具体如图 6-8 所示。



图 6-8　Kettle 检索字段

在图 6-8 中选择"元数据"选项卡，切换到"元数据"选项卡界面，如图 6-9 所示。



图 6-9　"元数据"选项卡界面

在图 6-9 中添加需要改变元数据的字段，将"字段选择"流中的字段进行一致性处理，即单击"获取改变的字段"按钮，获取要改变的字段，并在 Binary to Normal 列的下拉框中选择"是"，使得 CSV 文件 company_a. csv 中数据的字段类型与数据表 company_b 中数据的字段类型一致，具体如图 6-10 所示。

在图 6-10 中单击"确定"按钮，完成"字段选择"控件的配置。

4. 配置"表输入"控件

双击图 6-3 中的"表输入"控件，进入"表输入"界面，如图 6-11 所示。

在图 6-11 中单击"新建"按钮，配置数据库连接，配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 6-12 所示。

在图 6-11 的 SQL 框中编写查询数据表 company_b 中数据的 SQL 语句，然后单击"预览"按钮，查看数据表 company_b 的数据是否成功从 MySQL 数据库中抽取到表输入流中，



图 6-10 “元数据”选项卡的配置



图 6-11 “表输入”界面

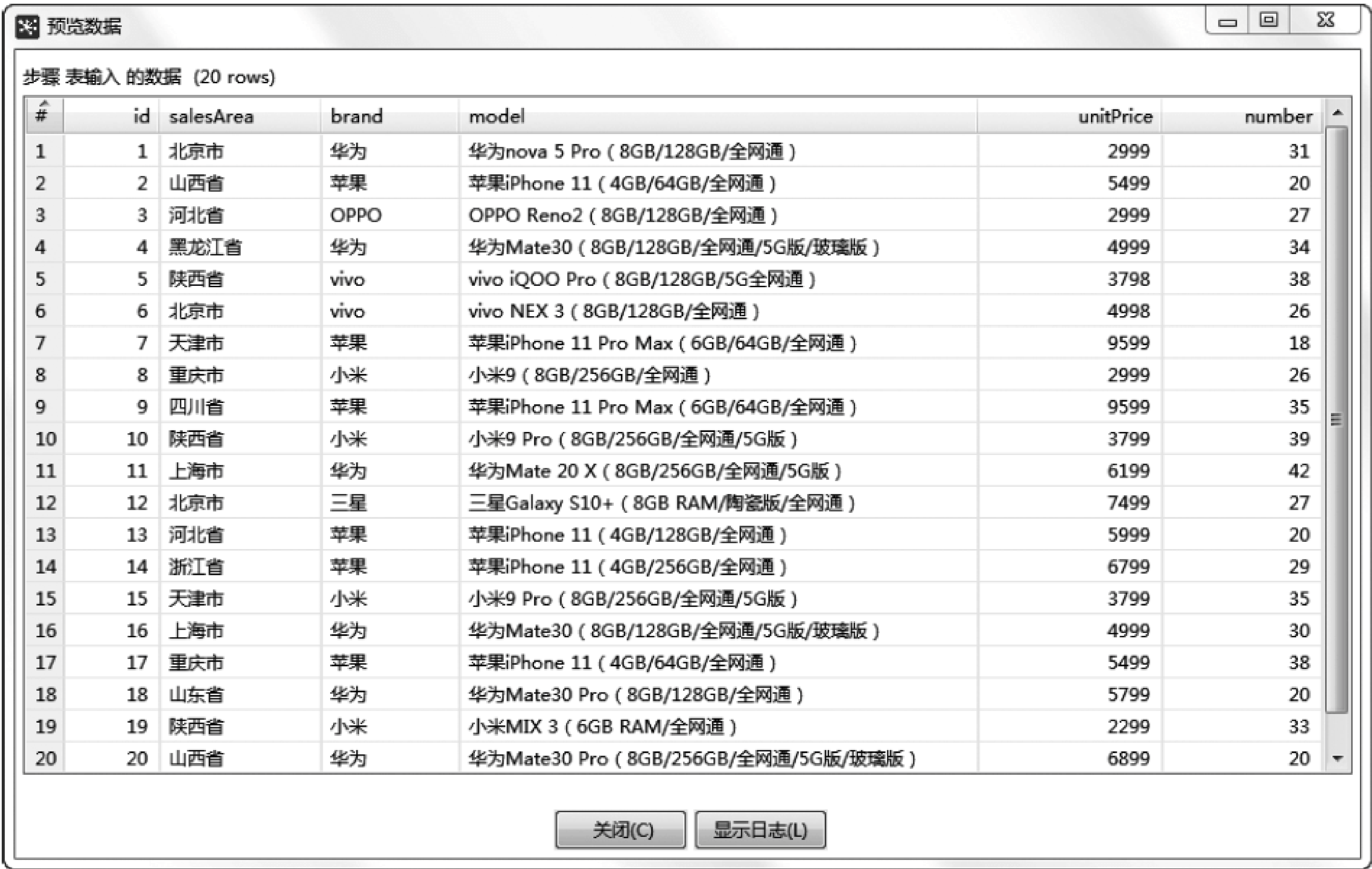


图 6-12 MySQL 数据库连接的配置

具体如图 6-13 和图 6-14 所示。



图 6-13　编写 SQL 语句



#	id	salesArea	brand	model	unitPrice	number
1	1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	2	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
3	3	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
4	4	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
5	5	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
6	6	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
7	7	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
8	8	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
9	9	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
10	10	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39
11	11	上海市	华为	华为Mate 20 X（8GB/256GB/全网通/5G版）	6199	42
12	12	北京市	三星	三星Galaxy S10+（8GB RAM/陶瓷版/全网通）	7499	27
13	13	河北省	苹果	苹果iPhone 11（4GB/128GB/全网通）	5999	20
14	14	浙江省	苹果	苹果iPhone 11（4GB/256GB/全网通）	6799	29
15	15	天津市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	35
16	16	上海市	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	30
17	17	重庆市	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	38
18	18	山东省	华为	华为Mate30 Pro（8GB/128GB/全网通）	5799	20
19	19	陕西省	小米	小米MIX 3（6GB RAM/全网通）	2299	33
20	20	山西省	华为	华为Mate30 Pro（8GB/256GB/全网通/5G版/玻璃版）	6899	20

图 6-14　预览数据

从图 6-14 中可以看出，数据表 company_b 的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

5. 配置“排序合并”控件

双击图 6-3 中的“排序合并”控件，进入“排序合并”界面，如图 6-15 所示。

在图 6-15 的字段框中指定按字段 id 进行升序排序，如图 6-16 所示。

在图 6-16 中单击“确定”按钮，完成“排序合并”控件的配置。



图 6-15 “排序合并”界面



图 6-16 “排序合并”控件的配置

6. 配置“增加序列”控件

双击图 6-3 中的“增加序列”控件，进入“增加序列”界面，如图 6-17 所示。



图 6-17 “增加序列”界面

在图 6-17 中的“值的名称”处指定要增加列的列名，这里使用默认的名称，即 valuename，其他配置项没有任何改变；单击“确定”按钮，完成“增加序列”控件的配置。

7. 配置“字段选择 2”控件

双击图 6-3 中的“字段选择 2”控件，进入“选择/改名值”界面，如图 6-18 所示。



图 6-18　“选择/改名值”界面

在图 6-18 中的“选择和修改”处选择和修改选择要输出的字段，具体如图 6-19 所示。



图 6-19　“字段选择 2”控件的配置

在图 6-19 中单击“确定”按钮，完成“字段选择 2”控件的配置。

8. 配置“表输出”控件

双击图 6-3 中的“表输出”控件，进入“表输出”界面，具体如图 6-20 所示。

在图 6-20 中单击“新建”按钮，配置数据库连接（所连接的数据库 transform 须提前创建，这里不赘述），配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-21 所示。

单击图 6-20 中目标表右侧的“浏览”按钮，指定输出目标表，即数据表 company（该表须提前创建，这里不演示）；勾选“指定数据库字段”复选框，用于将数据表 company 的字段与“字段选择 2”控件输出流中的字段进行匹配，如图 6-22 所示。

单击图 6-22 中的“数据库字段”选项卡，具体如图 6-23 所示。

在图 6-23 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，具体如图 6-24 所示。

在图 6-24 中依次选中“源字段”中的字段和“目标字段”对应的字段，再单击 Add 按钮，



图 6-20 “表输出”界面

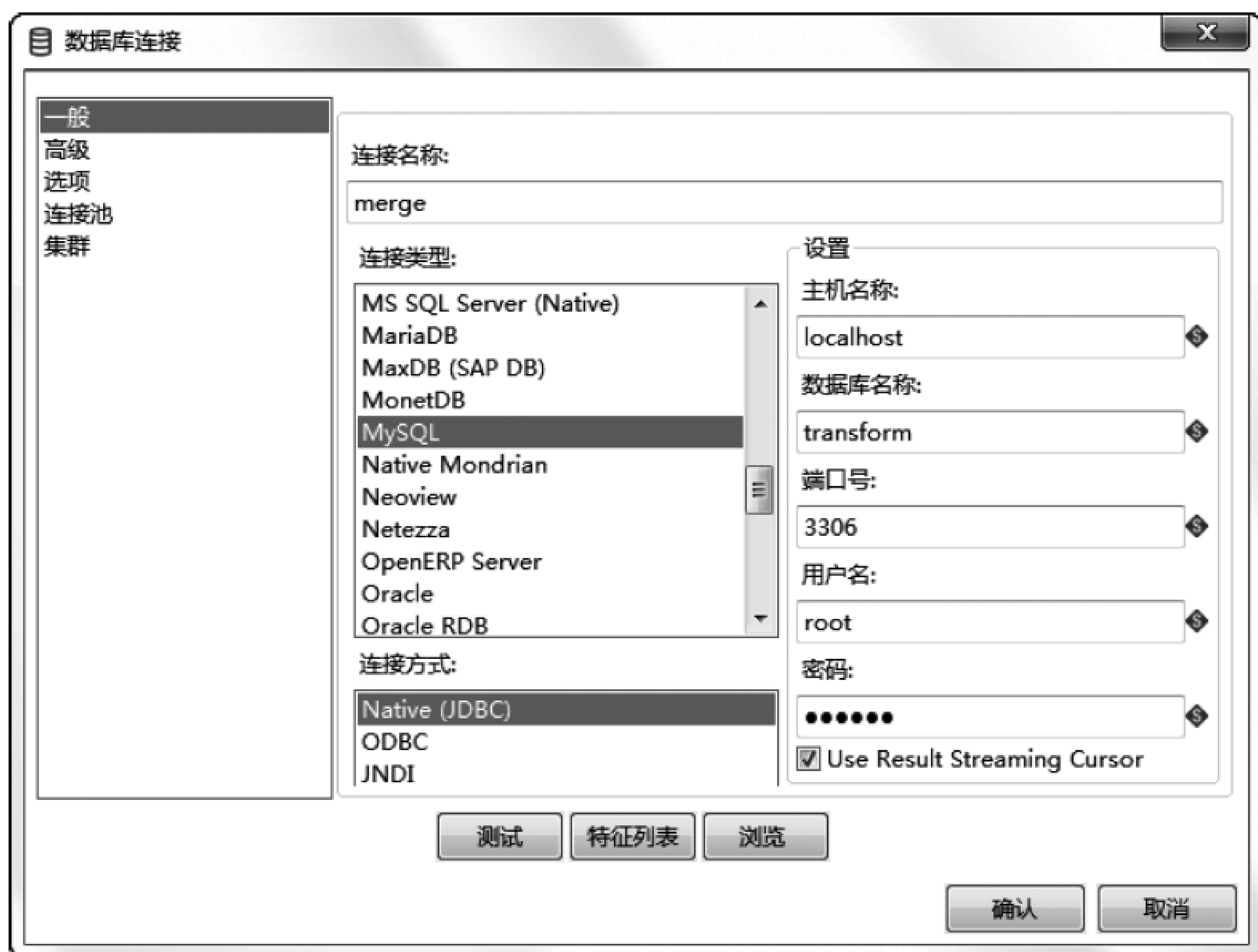


图 6-21 MySQL 数据库连接的配置

将一对映射字段添加至“映射”选项框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 6-25 所示。

在图 6-25 中单击“确定”按钮，完成“源字段”与“目标字段”的映射匹配。“表输出”控件配置的效果图如图 6-26 所示。

在图 6-26 中单击“确定”按钮，完成“表输出”控件的配置。

9. 运行转换 company_merge

单击转换工作区顶部的▷按钮，运行创建的转换 company_merge，实现将 A 公司和 B 公司的手机日销售数据合并到一个数据源(即数据表 company)中，具体如图 6-27 所示。



图 6-22　指定输出目标表和勾选“指定数据库字段”复选框



图 6-23　“数据库字段”选项卡



图 6-24　“映射匹配”对话框

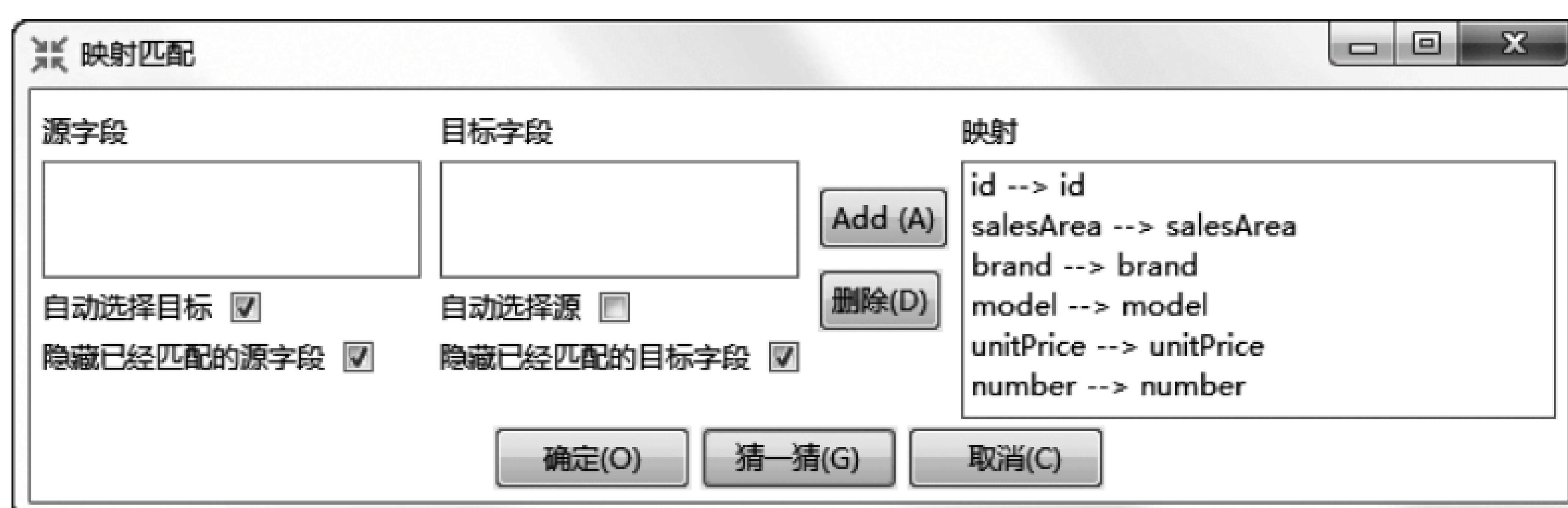


图 6-25 设置"映射匹配"



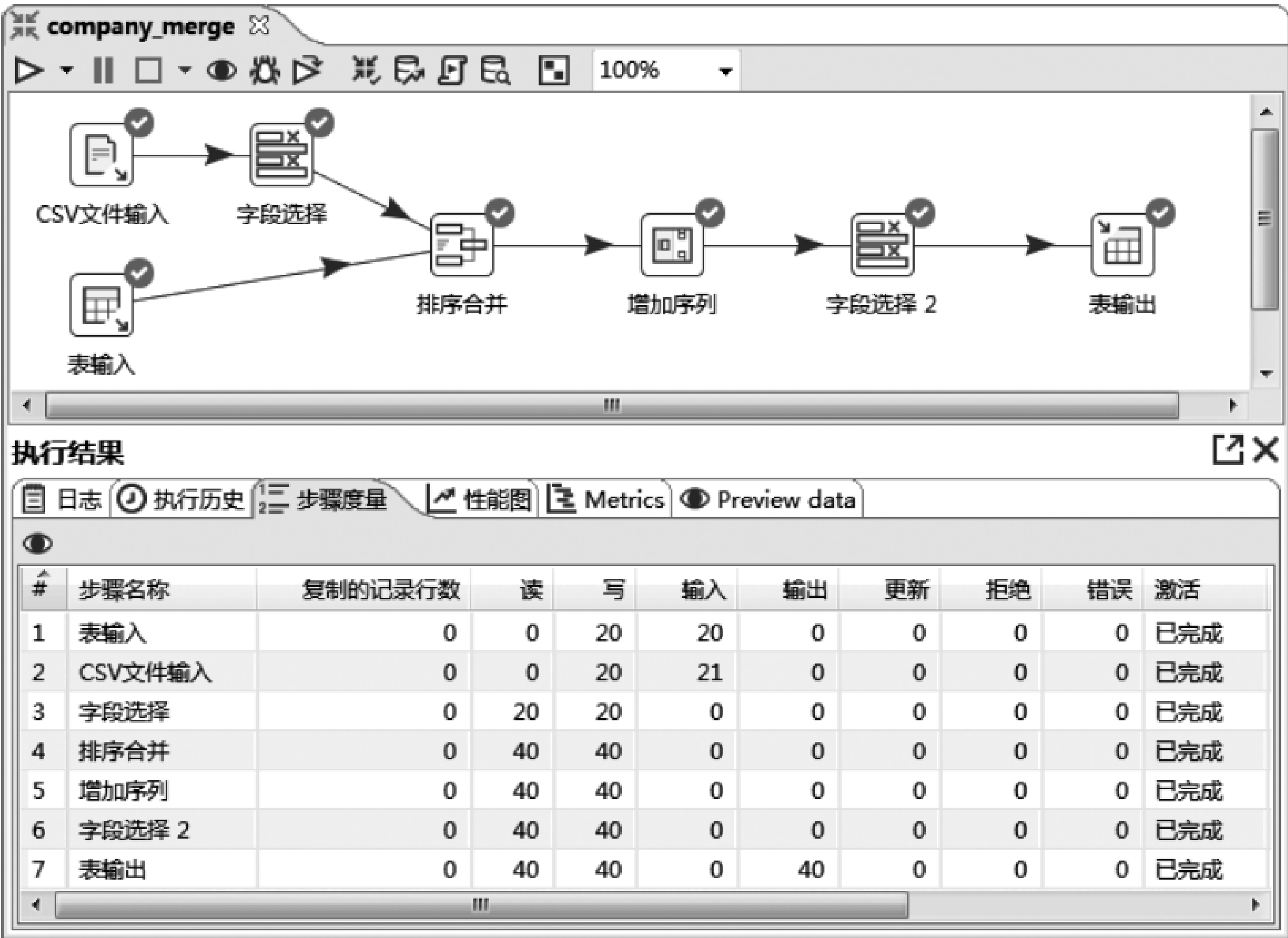
图 6-26 "表输出"控件配置的效果图

从图 6-27 中执行结果的"步骤度量"可以看出,"表输入"控件输入 20 条数据并写入该控件;"CSV 文件输入"控件输入 21 条数据并写入该控件 20 条数据(1 条表头数据除外);"字段选择"控件从"CSV 文件输入"中读取 20 条数据并写入该控件;"排序合并"控件从"表输入"控件和"字段选择"控件分别读取 20 条数据,并写入该控件;"增加序列"控件从"排序合并"控件中读取 40 条数据,并写入该控件;"字段选择 2"控件从"增加序列"控件中读取 40 条数据并写入该控件;"表输出"控件从"字段选择 2"控件中读取 40 条数据并写入该控件,最终进行输出。

10. 查看数据表 company 中的数据

通过 SQLyog 工具,查看数据表 company 是否已成功插入 40 条数据,查看结果如图 6-28 所示(只展示部分数据)。

从图 6-28 中可以看出,数据表 company 中插入了 40 条数据,说明成功实现了将 A 公司和 B 公司的手机日销售数据合并到一个数据源(即数据表 company)中。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	表输入	0	0	20	20	0	0	0	0	已完成
2	CSV文件输入	0	0	20	21	0	0	0	0	已完成
3	字段选择	0	20	20	0	0	0	0	0	已完成
4	排序合并	0	40	40	0	0	0	0	0	已完成
5	增加序列	0	40	40	0	0	0	0	0	已完成
6	字段选择 2	0	40	40	0	0	0	0	0	已完成
7	表输出	0	40	40	0	40	0	0	0	已完成

图 6-27　运行转换 company_merge

1 信息　2 表数据　3 信息

限制行　第一行：0　行数：1000

id	salesArea	brand	model	unitPrice	number
1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	石家庄市	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	大连市	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	秦皇岛市	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	青岛市	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	西安市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	大同市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9699	38
20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39
21	上海市	华为	华为Mate 20 X（8GB/256GB/全网通/5G版）	6199	42
22	哈尔滨市	OPPO	OPPO R17（8GB RAM/全网通）	2499	29
23	深圳市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	36
24	北京市	三星	三星Galaxy S10+（8GB RAM/陶瓷版/全网通）	7499	27
25	河北省	苹果	苹果iPhone 11（4GB/128GB/全网通）	5999	20
26	西安市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	15
27	太原市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	24
28	浙江省	苹果	苹果iPhone 11（4GB/256GB/全网通）	6799	29
29	天津市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	35
30	天津市	三星	三星GALAXY Note 10（8GB/256GB/全网通）	6599	23

数据库： transform　表格： company

图 6-28　数据表 company

6.2 不一致数据转换

不一致数据转换主要是将不同业务系统中的相同类型的数据进行统一。例如,同一供应商在结算系统中的编码是 XX0001,而在 CRM(客户关系管理)系统中的编码是 YY0001,这时就需要将这两个业务系统中的数据抽取过来进行统一转换,转换成同一个编码。

A 公司和 B 公司销售的手机均从同一个供货商手里采购,因此,同一品牌型号的手机,售价也应相同。但是,数据表 company 中存在同一品牌型号的手机,售价却不同,具体如图 6-29 所示。

id	salesArea	brand	model	unitPrice	number
2	北京市	苹果	苹果iPhone 11 Pro (6GB/64GB/全网通)	8699	50
3	上海市	苹果	苹果iPhone 11 Pro (6GB/256GB/全网通)	9999	20
4	山西省	苹果	苹果iPhone 11 (4GB/64GB/全网通)	5499	20
5	河北省	OPPO	OPPO Reno2 (8GB/128GB/全网通)	2999	27
6	北京市	三星	三星GALAXY Note 10+ (12GB/256GB/全网通/5G版)	7999	32
7	石家庄市	OPPO	OPPO Reno2 (8GB/128GB/全网通)	2999	20
8	黑龙江省	华为	华为Mate30 (8GB/128GB/全网通/5G版/玻璃版)	4999	34
9	陕西省	vivo	vivo iQOO Pro (8GB/128GB/5G全网通)	3798	38
10	大连市	OPPO	OPPO Reno 10倍变焦版 (8GB/256GB/全网通)	4299	35
11	秦皇岛市	vivo	vivo NEX 3 (8GB/256GB/全网通/5G版)	5698	26
12	北京市	vivo	vivo NEX 3 (8GB/128GB/全网通)	4998	26
13	天津市	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9599	18
14	青岛市	vivo	vivo iQOO Pro (8GB/128GB/5G全网通)	3798	25
15	西安市	华为	华为P30 Pro (8GB/128GB/全网通)	4988	30
16	重庆市	小米	小米9 (8GB/256GB/全网通)	2999	26
17	四川省	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9599	35
18	大同市	小米	小米9 Pro (8GB/256GB/全网通/5G版)	3799	22
19	重庆市	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9699	38
20	陕西省	小米	小米9 Pro (8GB/256GB/全网通/5G版)	3799	39
21	上海市	华为	华为Mate 20 X (8GB/256GB/全网通/5G版)	6199	42
22	哈尔滨市	OPPO	OPPO R17 (8GB RAM/全网通)	2499	29
23	深圳市	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9599	36
24	北京市	三星	三星Galaxy S10+ (8GB RAM/陶瓷版/全网通)	7499	27
25	河北省	苹果	苹果iPhone 11 (4GB/128GB/全网通)	5999	20
26	西安市	苹果	苹果iPhone 11 Pro (6GB/256GB/全网通)	9999	15
27	太原市	华为	华为P30 Pro (8GB/128GB/全网通)	4988	24
28	浙江省	苹果	苹果iPhone 11 (4GB/256GB/全网通)	6799	29
29	天津市	小米	小米9 Pro (8GB/256GB/全网通/5G版)	3799	35
30	天津市	三星	三星GALAXY Note 10 (8GB/256GB/全网通)	6599	23

图 6-29 数据表 company 中的不一致数据

从图 6-29 中可以看出,标记的 4 条数据中,字段 brand 和 model 均指向同一品牌和型号,而 id 为 19 的这条数据价格字段(unitPrice)与其他 3 条数据的价格字段不同。

下面通过 Kettle 工具对数据表 company 进行不一致数据的转换操作,即通过与供货商提供的标准价格表进行比较,得出不一致数据,从而进行修改,最终输出到数据表 company 中,具体实现步骤如下。

1. 打开 Kettle 工具,创建转换

使用 Kettle 工具创建转换 inconsistent,并添加"表输入"控件、"字段选择"控件、"记录集连接"控件、"插入/更新"控件以及 Hop 跳连接线,具体效果如图 6-30 所示。

2. 配置"表输入"控件

双击图 6-30 中的"表输入"控件,进入"表输入"界面,如图 6-31 所示。

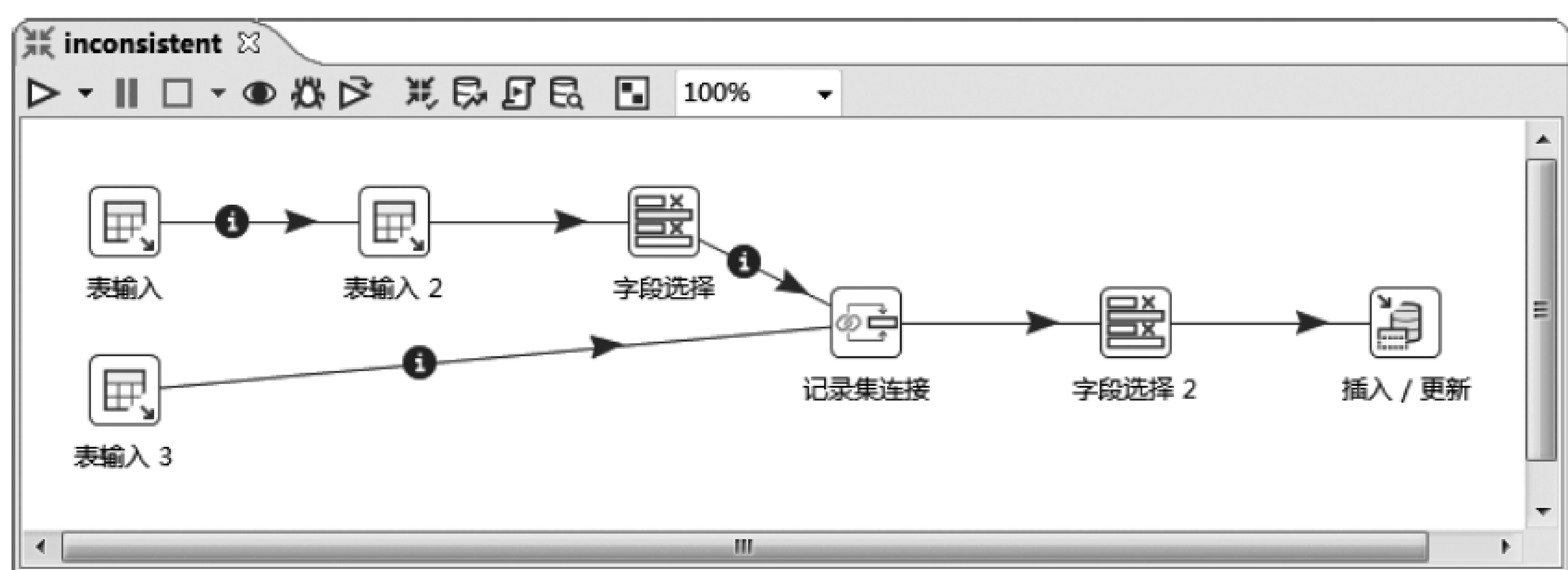


图 6-30　创建转换 inconsistent



图 6-31　“表输入”界面

在图 6-31 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-32 所示。

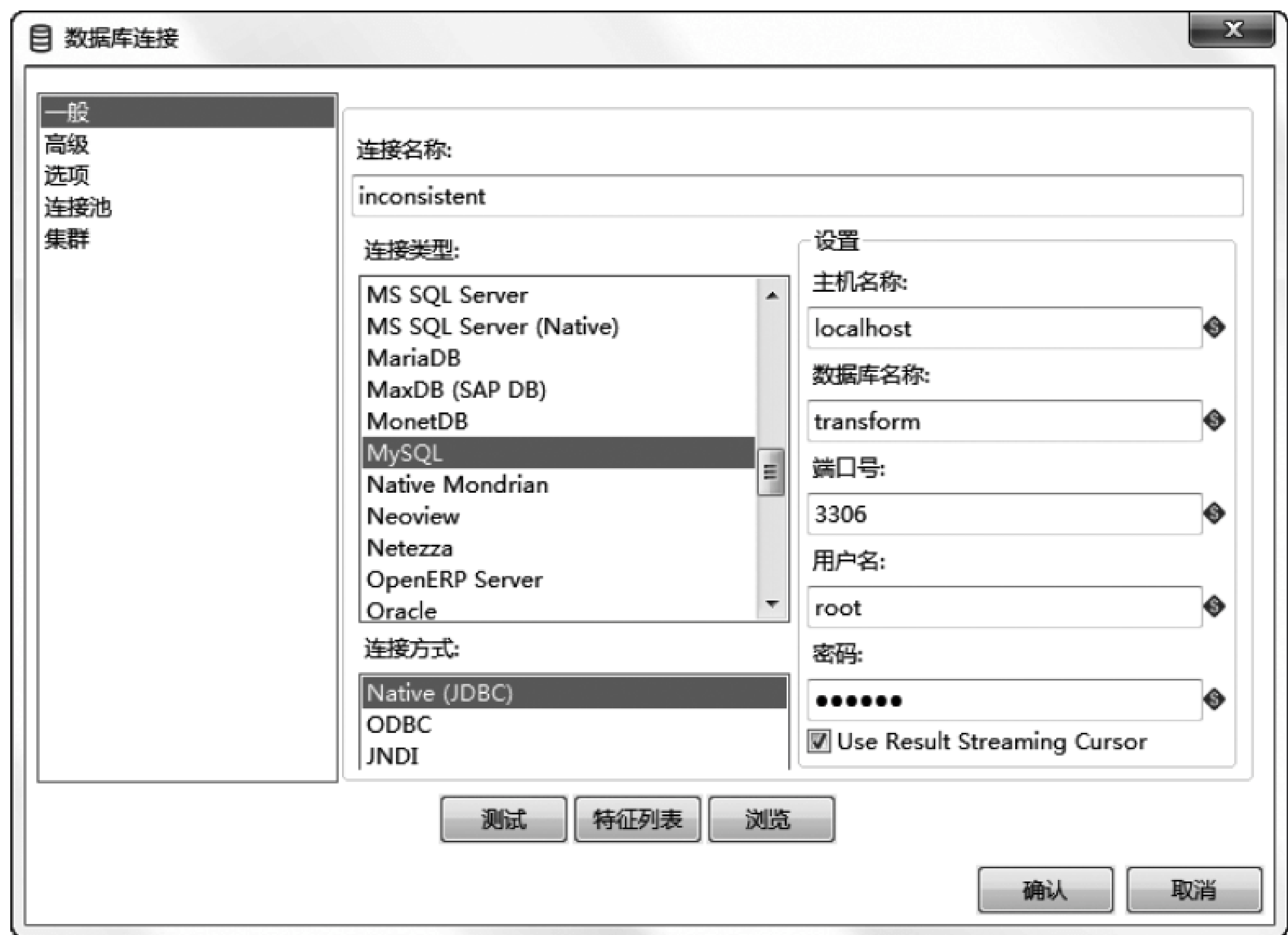


图 6-32　MySQL 数据库连接的配置

在图 6-31 的 SQL 框中编写查询数据表 company 中品牌、型号不一致数据的 SQL 语句，然后单击“预览”按钮，查看数据表 company 中品牌、型号不一致数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-33 和图 6-34 所示。



图 6-33 编写 SQL 语句



图 6-34 预览数据

从图 6-34 中可以看出，数据表 company 中品牌、型号不一致数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。需要注意的是，其中数据表 suppliers 为供应商出售手机的价格清单表，本书涉及的文件会作为素材提供给读者。

3. 配置“表输入 2”控件

双击图 6-30 中的“表输入 2”控件，进入“表输入”界面，如图 6-35 所示。

在图 6-35 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-36 所示。

在图 6-35 的 SQL 框中编写 SQL 语句，查询出不一致数据在数据表 suppliers 中的全部信息；在“从步骤插入数据”后的下拉列表中选择“表输入”控件；勾选“执行每一行”复选框，用于将“表输入”控件流中的数据放入 SQL 语句对应的参数位置，通过条件查询与数据表 suppliers 中的数据进行匹配，具体配置如图 6-37 所示。需要注意的是，“表输入”控件查询



图 6-35　“表输入”界面

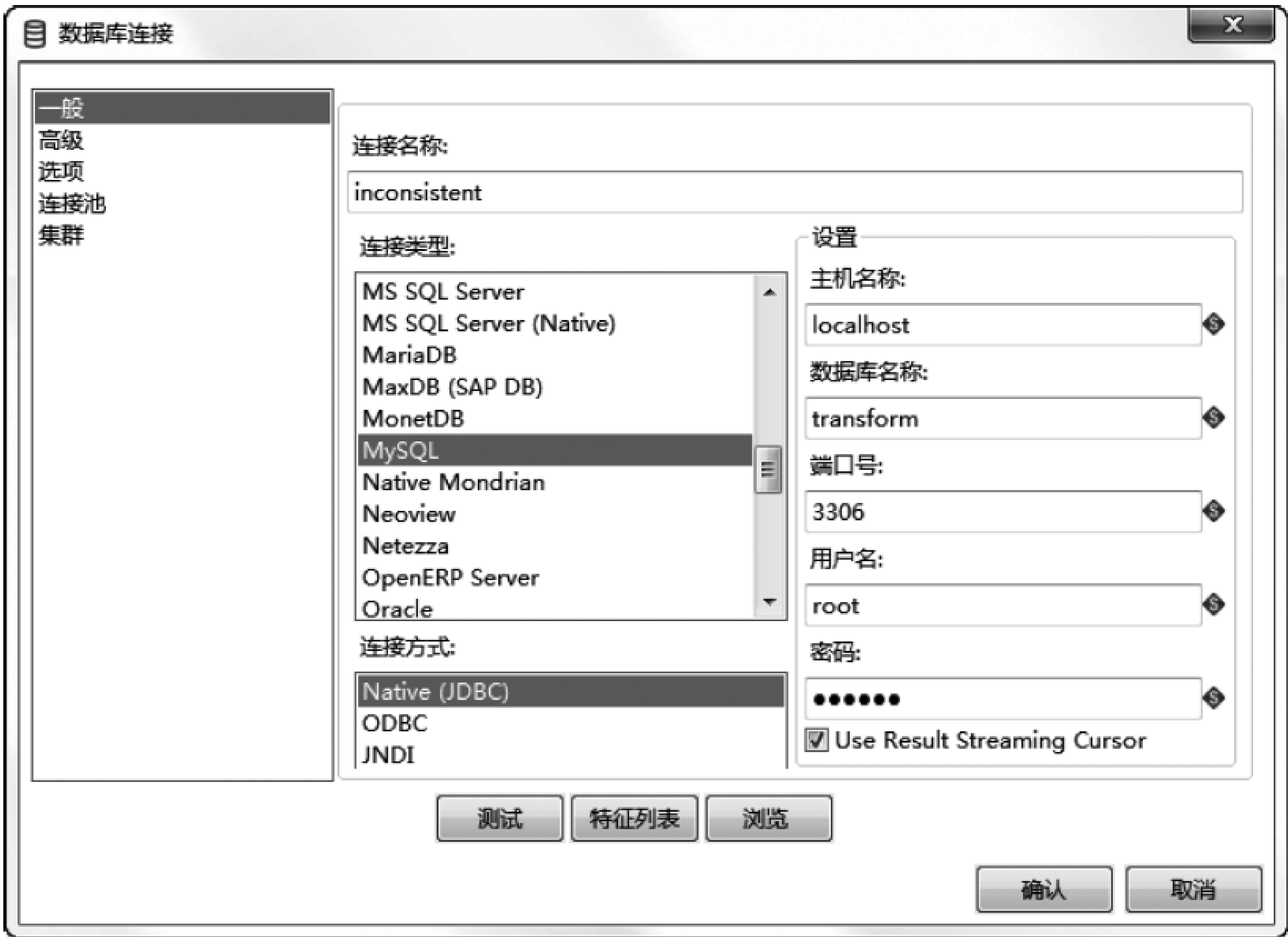


图 6-36　MySQL 数据库连接的配置

字段的顺序要与本控件内 SQL 语句的参数对应。

在图 6-37 中单击“确定”按钮，完成“表输入 2”控件的配置。

4. 配置“字段选择”控件

双击图 6-30 中的“字段选择”控件，进入“选择/改名值”界面，如图 6-38 所示。

在图 6-38 中“选择和修改”选项卡的“字段”处添加“表输入 2”控件流中的所有数据字段，并将字段 unitPrice 改名成 unitPrice1，具体如图 6-39 所示。

在图 6-39 中选择“移除”选项卡，切换到“移除”选项卡界面，如图 6-40 所示。

在图 6-40 中添加需要移除的字段，这里添加的是字段 id，由于后续操作不需要字段 id，



图 6-37 编写 SQL 语句



图 6-38 “选择/改名值”界面



图 6-39 Kettle 检索字段

因此在此进行移除，具体如图 6-41 所示。

在图 6-41 中单击“确定”按钮，完成“字段选择”控件的配置。

5. 配置“表输入 3”控件

双击图 6-30 中的“表输入 3”控件，进入“表输入”界面，如图 6-42 所示。



图 6-40　“移除”选项卡界面



图 6-41　移除 id 字段



图 6-42　“表输入”界面

在图 6-42 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-43 所示。

在图 6-42 的 SQL 框中编写查询与标准价格表(供货商提供的价格表)中品牌、型号不一致数据的 SQL 语句，然后单击“预览”按钮，查看数据表 company 中品牌、型号相同，价格不同的不一致数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-44 和图 6-45 所示。

从图 6-45 中可以看出，数据表 company 中品牌、型号相同，价格不同的不一致数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入 3”控

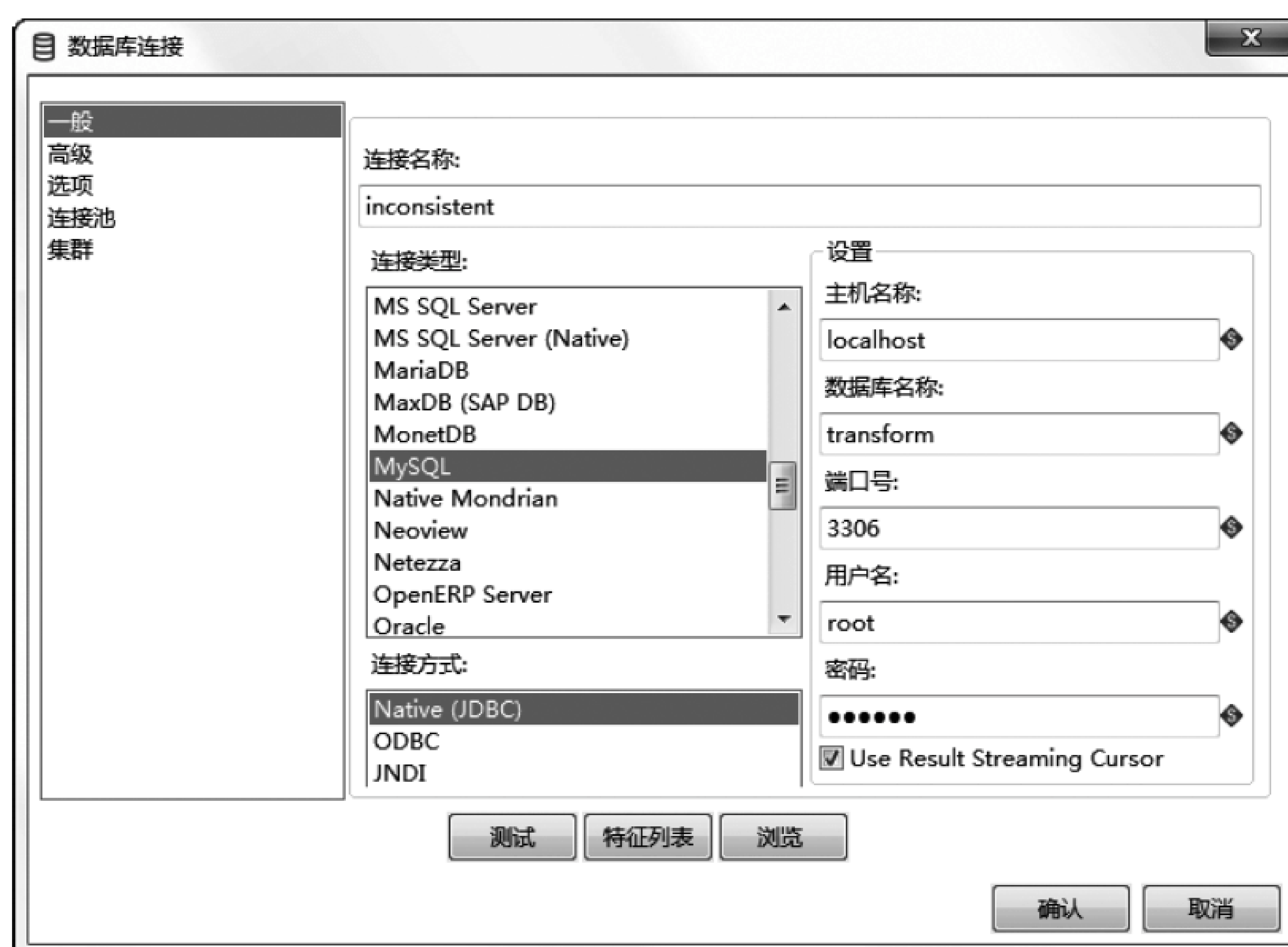


图 6-43　MySQL 数据库连接的配置



图 6-44　编写 SQL 语句



#	id	brand	model	unitPrice
1	19	苹果	苹果iPhone 11 Pro Max (6GB/64GB/全网通)	9699

图 6-45　预览数据

件的配置。

6. 配置“记录集连接”控件

双击图 6-30 中的“记录集连接”控件，进入“合并排序”界面，如图 6-46 所示。



图 6-46　“合并排序”界面

在图 6-46 中“第一个步骤”后的下拉列表中选择“字段选择”控件，在“第二个步骤”后的下拉列表中选择“表输入 3”控件，用于将“字段选择”控件流中的数据与“表输入 3”控件流中的数据进行合并连接；在“第一个步骤的连接字段”和“第二个步骤的连接字段”处添加连接字段，这里添加的连接字段是 brand、model，用于将“字段选择”控件流中的字段 brand、model 与“表输入 3”控件流中的字段 brand、model 进行连接；在“连接类型”后的下拉列表中选择连接类型，这里选择的是 RIGHT OUTER，即右外连接，具体如图 6-47 所示。



图 6-47　配置“记录集连接”控件

在图 6-47 中单击“确定”按钮，完成“记录集连接”控件的配置。

7. 配置“字段选择 2”控件

双击图 6-30 中的“字段选择 2”控件，进入“选择/改名值”界面，如图 6-48 所示。

在图 6-48 中“选择和修改” 选项卡的字段名称处填写 id 和 unitPrice1，用于在“插入/更新”控件中通过唯一字段 id 修改对应的价格字段 unitPrice1 内容，具体如图 6-49 所示。



图 6-48 “选择/改名值”界面



图 6-49 “字段选择 2”控件的配置

在图 6-49 中单击“确定”按钮，完成“字段选择 2”控件的配置。

8. 配置“插入/更新”控件

双击图 6-30 中的“插入/更新”控件，进入“插入/更新”界面，如图 6-50 所示。



图 6-50 “插入/更新”界面

在图 6-50 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-51 所示。

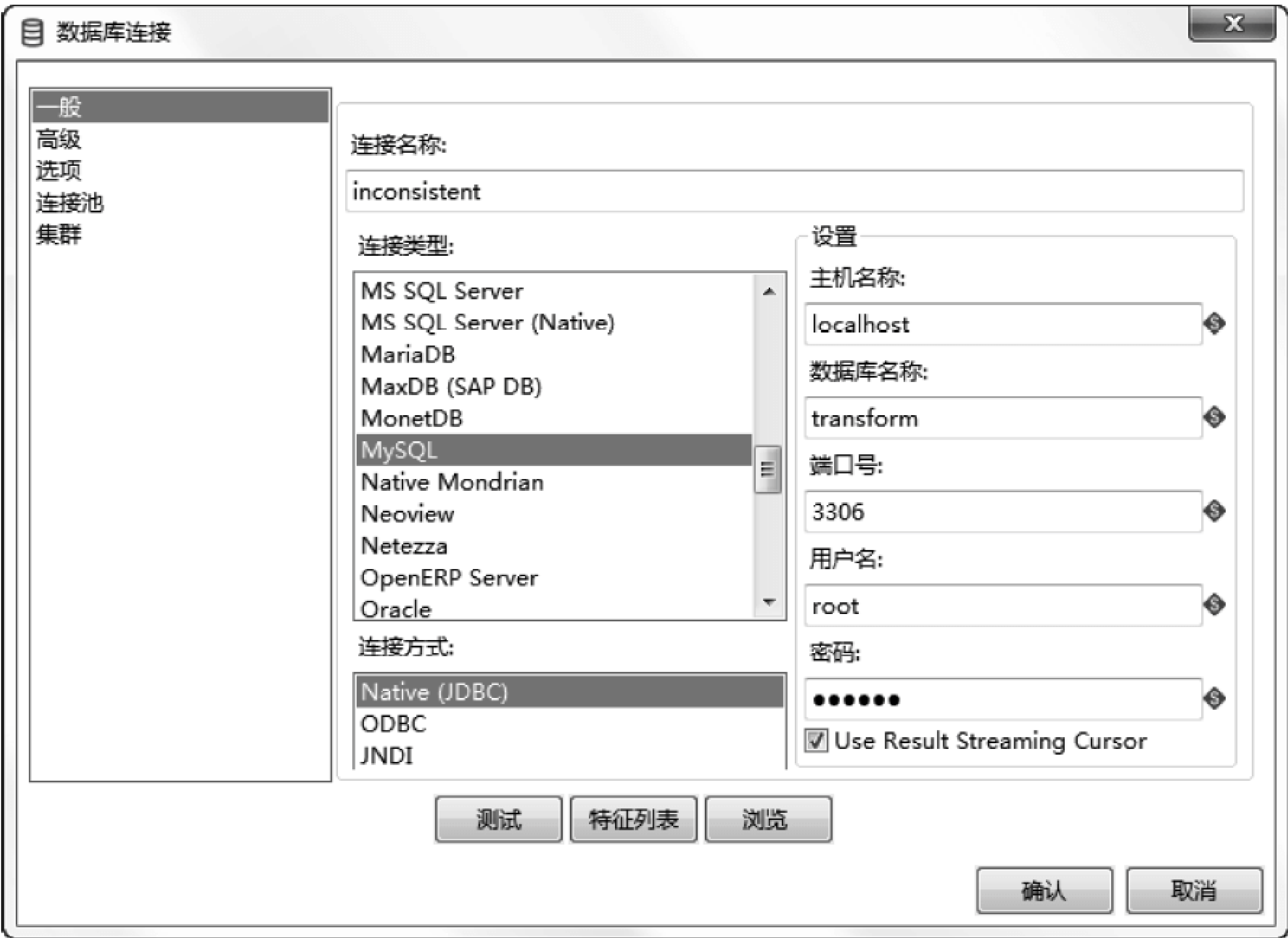


图 6-51　MySQL 数据库连接的配置

单击图 6-50 中目标表右侧的“浏览”按钮，弹出“数据库浏览器”界面，选择目标表 company；单击图 6-50 中的“获取字段”按钮，用来指定查询数据需要的关键字，也可通过手动输入，指定查询数据需要的关键字，这里选择的是数据表 company 中的 id 字段和输入流里的 id 字段；单击“获取和更新字段”按钮，用来指定需要更新的字段，具体如图 6-52 所示。

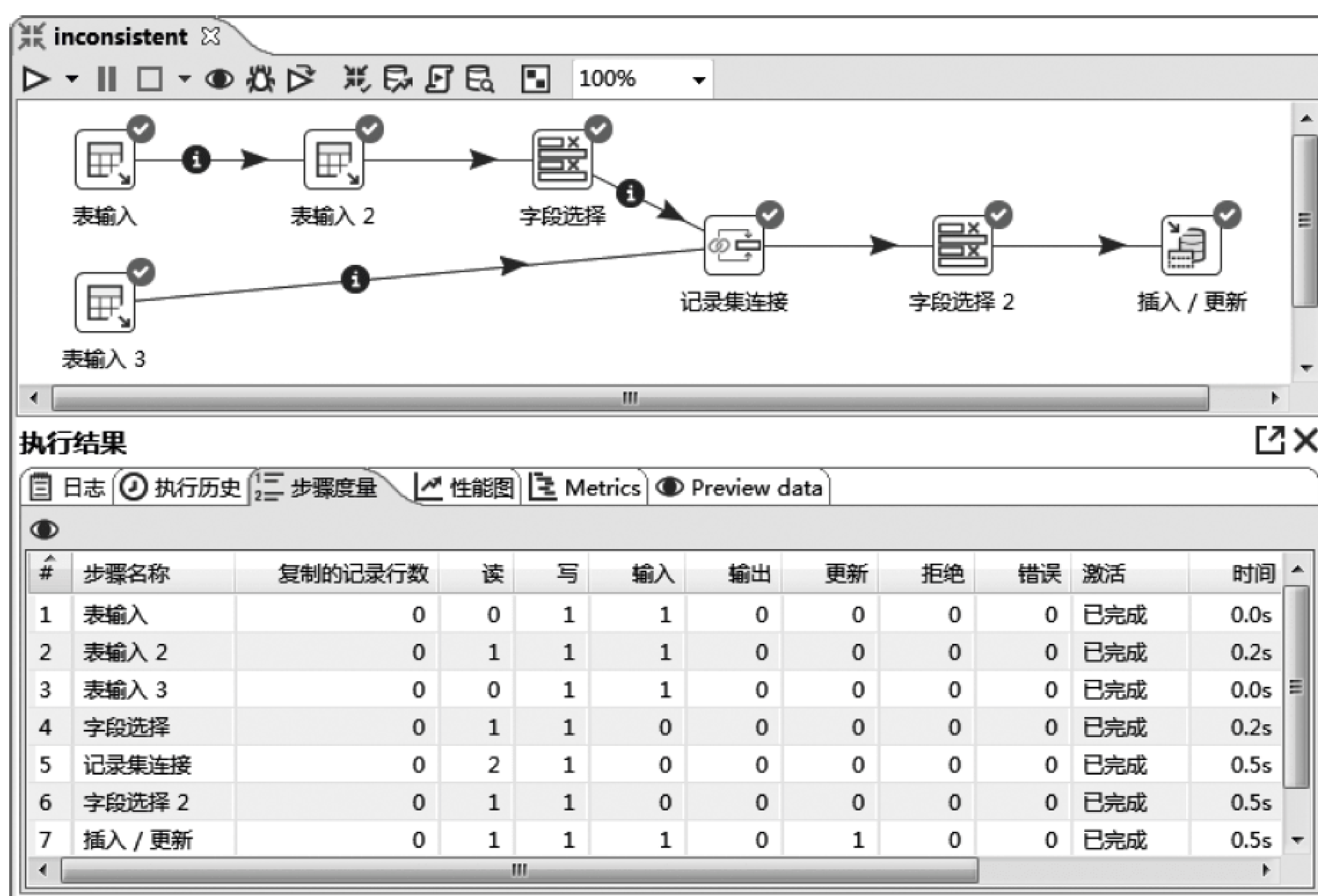


图 6-52　“插入/更新”控件的配置

在图 6-52 中单击“确定”按钮，完成“插入/更新”控件的配置。

9. 运行转换 inconsistent

单击转换工作区顶部的▷按钮，运行创建的转换 inconsistent，实现将数据表 company 中品牌、型号相同，价格不同的不一致数据修改成品牌、型号、价格均相同的一致数据，具体如图 6-53 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活	时间
1	表输入	0	0	1	1	0	0	0	0	已完成	0.0s
2	表输入 2	0	1	1	1	0	0	0	0	已完成	0.2s
3	表输入 3	0	0	1	1	0	0	0	0	已完成	0.0s
4	字段选择	0	1	1	0	0	0	0	0	已完成	0.2s
5	记录集连接	0	2	1	0	0	0	0	0	已完成	0.5s
6	字段选择 2	0	1	1	0	0	0	0	0	已完成	0.5s
7	插入 / 更新	0	1	1	1	0	1	0	0	已完成	0.5s

图 6-53　运行转换 inconsistent

从图 6-53 中执行结果的“步骤度量”可以看出，“表输入”控件输入 1 条数据并写入该控件；“表输入 2”控件读取“表输入 1”控件的 1 条数据，作为“表输入 2”控件的输入，并写入该控件；“表输入 3”控件输入 1 条数据并写入该控件；“字段选择”控件从“表输入 2”控件中读取 1 条数据并写入该控件；“记录集连接”控件从“字段选择”和“表输入 3”控件中各读取 1 条数据，并将合并后的 1 条数据写入该控件；“字段选择 2”控件从“记录集连接”控件中读取 1 条数据并写入该控件；“插入/更新”控件读取“字段选择 2”控件的 1 条数据，作为“插入/更新”控件的输入并写入该控件完成对数据库 1 条数据的更新操作。

10. 查看数据表 company 中的数据

通过 SQLyog 工具，查看数据表 company 是否已成功将数据表 company 中品牌、型号相同，价格不同的数据修改成品牌、型号、价格均相同的数据，查看结果如图 6-54 所示（只展示部分数据）。

从图 6-54 中可以看出，数据表 company 中字段 brand 为“苹果”、model 为“苹果 iPhone 11 Pro Max（6GB/64GB/全网通）”的 unitPrice 均为 9599，说明成功实现了将数据表 company 中品牌、型号相同，价格不同的数据修改成品牌、型号、价格均相同的数据。

id	salesArea	brand	model	unitPrice	number
1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	石家庄市	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	大连市	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	秦皇岛市	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	青岛市	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	西安市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	大同市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39
21	上海市	华为	华为Mate 20 X（8GB/256GB/全网通/5G版）	6199	42
22	哈尔滨市	OPPO	OPPO R17（8GB RAM/全网通）	2499	29
23	深圳市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	36
24	北京市	三星	三星Galaxy S10+（8GB RAM/陶瓷版/全网通）	7499	27
25	河北省	苹果	苹果iPhone 11（4GB/128GB/全网通）	5999	20
26	西安市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	15
27	太原市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	24
28	浙江省	苹果	苹果iPhone 11（4GB/256GB/全网通）	6799	29
29	天津市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	35
30	天津市	三星	三星GALAXY Note 10（8GB/256GB/全网通）	6599	23

图 6-54　数据表 company

6.3　数据粒度的转换

业务系统一般存储非常明细的数据，而数据仓库中数据是用来分析的，不需要非常明细的数据。一般情况下会将业务系统数据按照数据仓库粒度进行聚合，这个过程被称为数据粒度的转换。例如，将城市转换成省份或者直辖市。

A 公司的日手机销售情况中的销售区域是市级，而 B 公司的日手机销售情况中的销售区域是省级，A 公司和 B 公司的日手机销售情况合并后存储在数据表 company 中，通过仔细观察数据表 company 中字段为 salesArea 的一列，发现该列既包含市级，也包含省级。数据表 company 中的部分数据内容如图 6-55 所示。

下面通过 Kettle 工具对数据表 company 进行数据粒度的转换操作，即将数据表 company 中字段为 salesArea 的数据都统一成省份级，并存储到新数据表 company_new 中，具体步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 granularity，并添加"表输入"控件、"字段选择"控件、"排序记录"控件、"记录集连接"控件、"过滤记录"控件、"空操作"控件、"表输出"控件以及 Hop 跳连接线，具体效果如图 6-56 所示。

2. 配置"表输入"控件

双击图 6-56 中的"表输入"控件，进入"表输入"界面，如图 6-57 所示。

id	salesArea	brand	model	unitPrice	number
1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	石家庄市	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	大连市	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	秦皇岛市	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	青岛市	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	西安市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	大同市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39

图 6-55 数据表 company

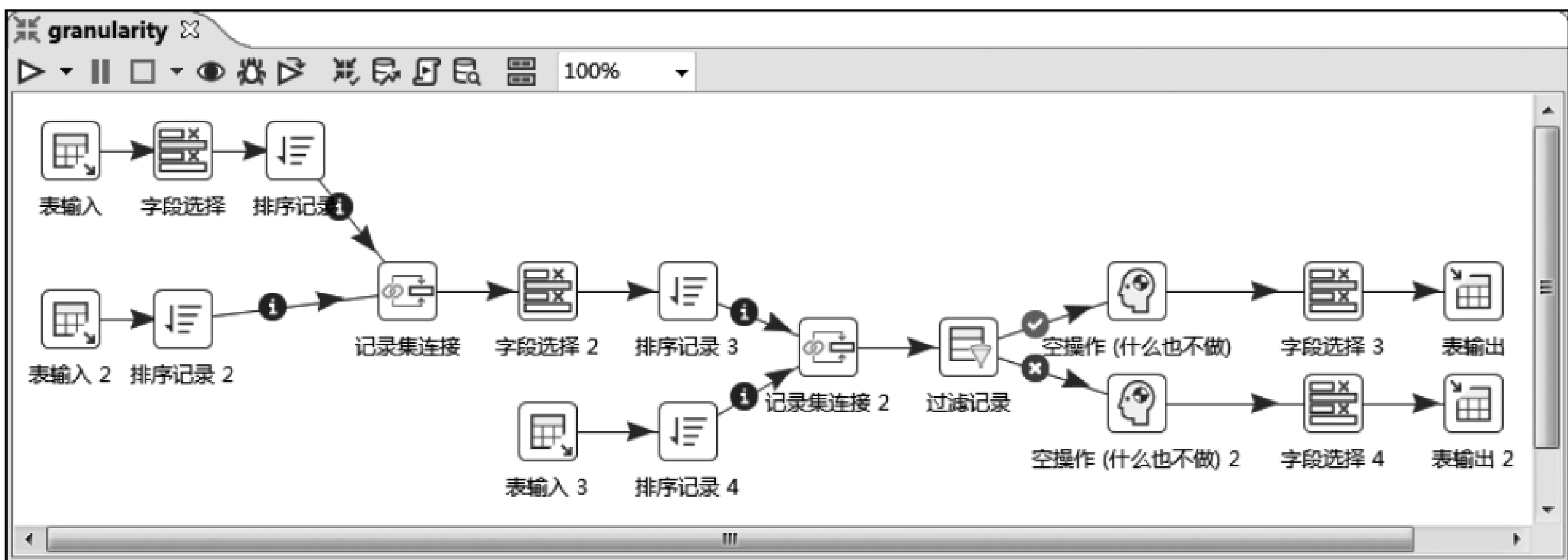


图 6-56 创建转换 granularity

表输入
步骤名称 表输入
数据库连接 编辑... 新建... Wizard...
SQL 获取SQL查询语句...
SELECT <values> FROM <table name> WHERE <conditions>
行1 列0
允许简易转换
替换 SQL 语句里的变量
从步骤插入数据
执行每一行?
记录数量限制 0
Help 确定(O) 预览(P) 取消(C)

图 6-57 “表输入”界面

在图 6-57 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-58 所示。



图 6-58　MySQL 数据库连接的配置

在图 6-57 的 SQL 框中编写获取数据表 company 中所有数据的 SQL 语句，然后单击“预览”按钮，查看数据表 company 中的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-59 和图 6-60 所示。



图 6-59　编写 SQL 语句

从图 6-60 中可以看出，数据表 company 中的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“字段选择”控件

双击图 6-56 中的“字段选择”控件，进入“选择/改名值”界面，如图 6-61 所示。



#	id	salesArea	brand	model	unitPrice	number
1	1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	7	石家庄市	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	10	大连市	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	11	秦皇岛市	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	14	青岛市	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	15	西安市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	18	大同市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39

图 6-60　预览数据



图 6-61　“选择/改名值”界面

在图 6-61 中“选择和修改”选项卡的“字段”处手动添加所需字段，这里添加字段 id 和 salesArea，具体如图 6-62 所示。

在图 6-62 中单击“确定”按钮，完成“字段选择”控件的配置。

4. 配置“排序记录”控件

双击图 6-56 中的“排序记录”控件，进入“排序记录”界面，如图 6-63 所示。

在图 6-63 的“字段”框中添加字段 salesArea，以该字段为基础对整体数据进行升序排序，具体如图 6-64 所示。

在图 6-64 中单击“确定”按钮，完成“排序记录”控件的配置。



图 6-62 添加所需字段



图 6-63 “排序记录”界面



图 6-64 配置“排序记录”控件

5. 配置“表输入 2”控件

双击图 6-56 中的“表输入 2”控件，进入“表输入”界面，如图 6-65 所示。



图 6-65 “表输入”界面

在图 6-65 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-66 所示。



图 6-66 MySQL 数据库连接的配置

在图 6-65 的 SQL 框中编写 SQL 语句，用于查询 city 数据表(注：该表须提前创建，读者执行本书提供的 data. sql 脚本文件可生成 city 数据表)中字段 city 和 pid 的数据，用于与数据表 company 中的字段 salesArea 进行合并连接，然后单击“预览”按钮，查看 city 数据表中的字段 city 和 pid 的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-67

和图 6-68 所示。



图 6-67　编写 SQL 语句

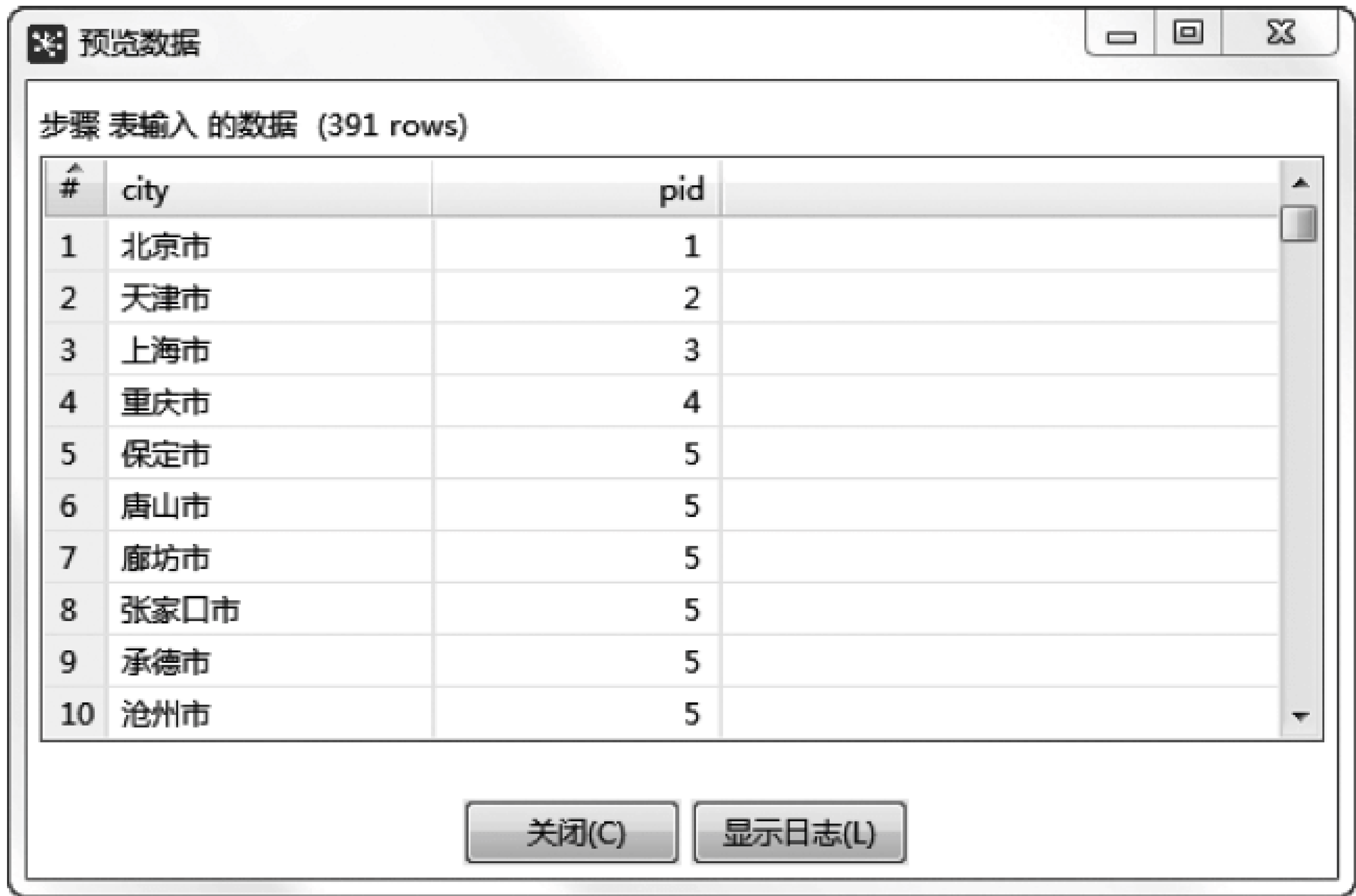


图 6-68　预览数据

从图 6-68 中可以看出，数据表 city 中的字段 city 和 pid 的数据已成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入 2”控件的配置。

6. 配置“排序记录 2”控件

双击图 6-56 中的“排序记录 2”控件，进入“排序记录”界面，具体如图 6-69 所示。

在图 6-69 的“字段”框中添加字段 city，以该字段为基础对整体数据进行升序排序，具体如图 6-70 所示。

在图 6-70 中单击“确定”按钮，完成“排序记录 2”控件的配置。

7. 配置“记录集连接”控件

双击图 6-56 中的“记录集连接”控件，进入“合并排序”界面，如图 6-71 所示。

在图 6-71 的“第一个步骤”后的下拉列表中选择“排序记录”控件，在“第二个步骤”后的

图 6-69 “排序记录”界面

图 6-70 配置“排序记录 2”控件

图 6-71 “合并排序”界面

下拉列表中选择“排序记录 2”控件;在“第一个步骤的连接字段”和“第二个步骤的连接字段”处分别添加连接字段,这里添加的连接字段分别是 salesArea 和 city,将“排序记录”控件流中的数据与“排序记录 2”控件流中的数据进行合并连接;在“连接类型”后的下拉列表中选择连接类型,这里选择的是 LEFT OUTER,即左外连接,具体如图 6-72 所示。



图 6-72　配置“记录集连接”控件

在图 6-72 中单击“确定”按钮,完成“记录集连接”控件的配置。

8. 配置“字段选择 2”控件

双击图 6-56 中的“字段选择 2”控件,进入“选择/改名值”界面,如图 6-73 所示。



图 6-73　“选择/改名值”界面

在图 6-73 中“选择和修改”选项卡的“字段”处手动添加所需字段,这里添加字段 id、pid 和 salesArea,具体如图 6-74 所示。

在图 6-74 中单击“确定”按钮,完成“字段选择 2”控件的配置。

9. 配置“排序记录 3”控件

双击图 6-56 中的“排序记录 3”控件,进入“排序记录”界面,具体如图 6-75 所示。

在图 6-75 的“字段”框中添加字段 pid,以该字段为基础对整体数据进行升序排序,具体如图 6-76 所示。

在图 6-76 中单击“确定”按钮,完成“排序记录 3”控件的配置。



图 6-74 “字段选择 2”控件的配置



图 6-75 “排序记录”界面



图 6-76 配置“排序记录 3”控件

10. 配置“表输入 3”控件

双击图 6-56 中的“表输入 3”控件，进入“表输入”界面，如图 6-77 所示。



图 6-77 “表输入”界面

在图 6-77 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-78 所示。



图 6-78 MySQL 数据库连接的配置

在图 6-77 的 SQL 框中编写 SQL，用于获取数据表 provincial 中字段为 pid 和 Provincial 的数据，后续用于与“排序记录”控件流中的字段数据进行合并连接，然后单击“预览”按钮，查看数据表 provincial 中字段为 pid 和 Provincial 的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-79 和图 6-80 所示。



图 6-79　编写 SQL 语句


预览数据

步骤 表输入 3 的数据 (34 rows)

#	pid	Provincial
1	1	北京市
2	2	天津市
3	3	上海市
4	4	重庆市
5	5	河北省
6	6	山西省
7	7	台湾省
8	8	辽宁省
9	9	吉林省
10	10	黑龙江省
11	11	江苏省
12	12	浙江省
13	13	安徽省
14	14	福建省
15	15	江西省

关闭(C)　显示日志(L)

图 6-80　预览数据

从图 6-80 中可以看出，数据表 provincial 中的字段 Provincial 和 pid 的数据已经成功从 MySQL 数据库中抽取到“表输入 3”控件的输入流中，单击“关闭”→“确定”按钮，完成“表输入 3”控件的配置。

11. 配置“排序记录 4”控件

双击图 6-56 中的“排序记录 4”控件，进入“排序记录”界面，具体如图 6-81 所示。

在图 6-81 的“字段”框中添加字段 pid，以该字段为基础对整体数据进行升序排序，具体如图 6-82 所示。

在图 6-82 中单击“确定”按钮，完成“排序记录 4”控件的配置。



图 6-81　“排序记录”界面



图 6-82　配置“排序记录 4”控件

12. 配置“记录集连接 2”控件

双击图 6-56 中的“记录集连接 2”控件，进入“合并排序”界面，如图 6-83 所示。



图 6-83　“合并排序”界面

在图 6-83 中“第一个步骤：”后的下拉列表中选择“排序记录 3”控件，在“第二个步骤：”后的下拉列表中选择“排序记录 4”控件；在“第一个步骤的连接字段”和“第二个步骤的连接字段”处分别添加连接字段，这里添加的连接字段分别是 pid 和 pid；在“连接类型”后的下拉列表中选择连接类型，这里选择的是 LEFT OUTER，即左外连接，具体如图 6-84 所示。



图 6-84　配置“记录集连接 2”控件

在图 6-84 中单击“确定”按钮，完成“记录集连接 2”控件的配置。

13．配置“过滤记录”控件

双击图 6-56 中的“过滤记录”控件，进入“过滤记录”界面，如图 6-85 所示。



图 6-85　“过滤记录”界面

在图 6-85 中的“条件”处设置过滤的条件，将字段为 Provincial 中值为 null 的数据过滤掉，具体如图 6-86 所示。

在图 6-86 中“发送 true 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)”，将 Provincial 字段值不为 null 的数据放在“空操作(什么也不做)”控件中；在“发送 false 数据给步骤：”后的下拉列表中选择“空操作(什么也不做)2”，将 Provincial 字段值为 null 的数据放在“空操作(什么也不做)2”控件中，具体如图 6-87 所示。

在图 6-87 中单击“确定”按钮，完成“过滤记录”控件的配置。

14．配置“字段选择 3”控件

双击图 6-56 中的“字段选择 3”控件，进入“选择/改名值”界面，如图 6-88 所示。



图 6-86　设置过滤条件



图 6-87　配置“发送 true/false 数据给相关步骤”



图 6-88　“选择/改名值”界面

在图 6-88 的“选择和修改”选项卡的“字段”处手动添加所需字段，这里添加字段 id 和 Provincial，用于后续进行输出，具体如图 6-89 所示。

在图 6-89 中单击“确定”按钮，完成“字段选择 3”控件的配置。

15. 配置“表输出”控件

双击图 6-56 中的“表输出”控件，进入“表输出”界面，具体如图 6-90 所示。

在图 6-90 中单击“新建”按钮，配置数据库连接(所连接的数据库 transform 须提前创建，这里不赘述)，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-91 所示。

单击图 6-90 中目标表右侧的“浏览”按钮，指定输出目标表，即数据表 temporary(该表



图 6-89 “字段选择 3”控件的配置



图 6-90 “表输出”界面



图 6-91 MySQL 数据库连接的配置

须提前创建，这里不演示）；勾选“指定数据库字段”复选框，用于将数据表 temporary 的字段与“字段选择 3”控件输出流中的字段进行匹配，如图 6-92 所示。



图 6-92　指定输出目标表和勾选“指定数据库字段”复选框

单击图 6-92 中的“数据库字段”选项卡，具体如图 6-93 所示。



图 6-93　“数据库字段”选项卡

在图 6-93 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，具体如图 6-94 所示。

在图 6-94 中依次选中“源字段”选项中的字段和“目标字段”中对应的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 6-95 所示。

在图 6-95 中单击“确定”按钮，完成“源字段”与“目标字段”的映射匹配。“表输出”控件配置的效果图如图 6-96 所示。

在图 6-96 中单击“确定”按钮，完成“表输出”控件的配置。



图 6-94 “映射匹配”对话框



图 6-95 设置映射匹配



图 6-96 “表输出”控件配置的效果图

16. 配置“字段选择 4”控件

双击图 6-56 中的“字段选择 4”控件，进入“选择/改名值”界面，如图 6-97 所示。

在图 6-97 的“选择和修改”选项卡的“字段”处手动添加所需字段，这里添加字段 id 和 salesArea，用于后续进行输出，具体如图 6-98 所示。

在图 6-98 中单击“确定”按钮，完成“字段选择 4”控件的配置。



图 6-97　“选择/改名值”界面



图 6-98　“字段选择 4”控件的配置

17. 配置“表输出 2”控件

双击图 6-56 中的“表输出 2”控件，进入“表输出”界面，具体如图 6-99 所示。



图 6-99　“表输出”界面

在图 6-99 中单击“新建”按钮，配置数据库连接(所连接的数据库 transform 须提前创建，这里不赘述)，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-100 所示。

单击图 6-99 中目标表右侧的“浏览”按钮，指定输出目标表，即数据表 temporary(该表



图 6-100 MySQL 数据库连接的配置

须提前创建,这里不演示);勾选“指定数据库字段”复选框,用于将数据表 temporary 的字段与“字段选择 4”控件输出流中的字段进行匹配,如图 6-101 所示。



图 6-101 指定输出目标表和勾选“指定数据库字段”复选框

单击图 6-101 中的“数据库字段”选项卡,具体如图 6-102 所示。

在图 6-102 中单击“输入字段映射”按钮,弹出“映射匹配”对话框,具体如图 6-103 所示。

在图 6-103 中依次选中“源字段”中的字段和“目标字段”中对应的字段,再单击 Add 按钮,将一对映射字段添加至“映射”框中,若“源字段”中的字段和“目标字段”中的字段相同,则可以单击“猜一猜”按钮,让 Kettle 自动实现映射,具体如图 6-104 所示。



图 6-102　"数据库字段"选项卡



图 6-103　"映射匹配"对话框



图 6-104　设置映射匹配

在图 6-104 中单击"确定"按钮，完成"源字段"与"目标字段"的映射匹配。"表输出 2"控件配置的效果图如图 6-105 所示。

在图 6-105 中单击"确定"按钮，完成"表输出 2"控件的配置。

18. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 granularity_merge，并添加"表输入"控件、"字段选择"控件、"排序记录"控件、"记录集连接"控件、"表输出"控件以及 Hop 跳连接线，具体效果如图 6-106 所示。



图 6-105 “表输出 2”控件配置的效果图

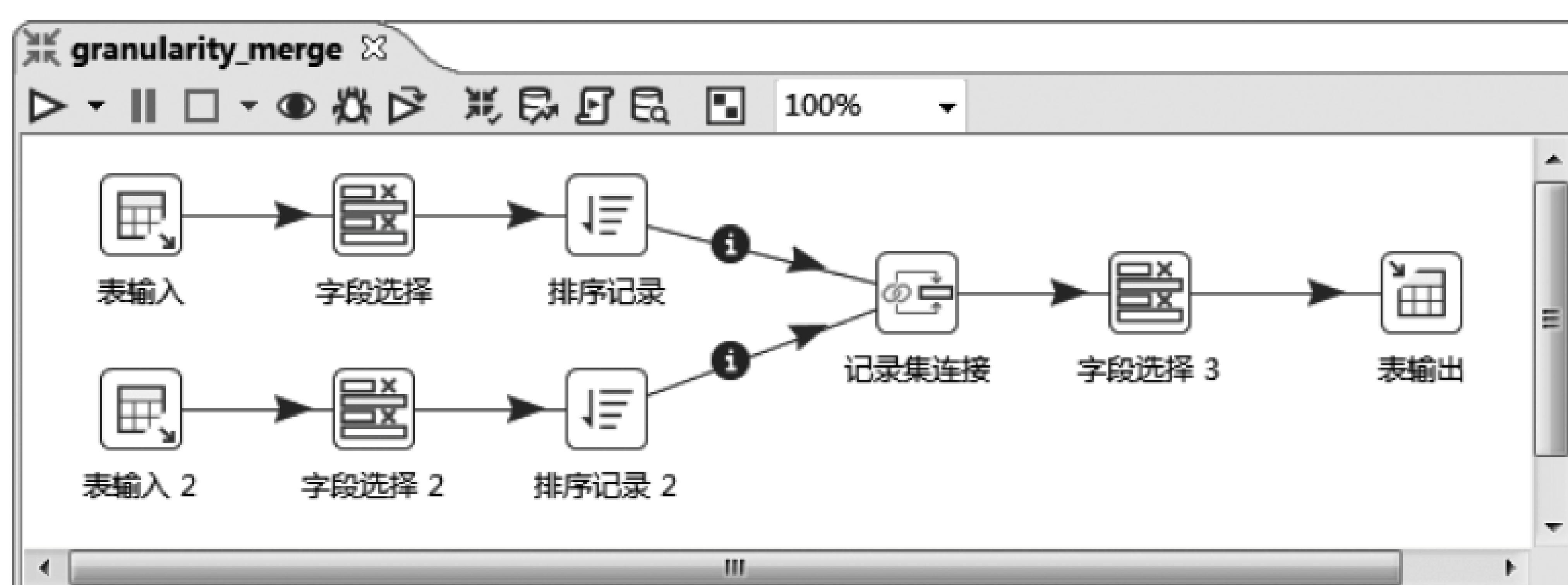


图 6-106 创建转换 granularity_merge

19. 配置“表输入”控件

双击图 6-106 中的“表输入”控件，进入“表输入”界面，如图 6-107 所示。



图 6-107 “表输入”界面

在图 6-107 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-108 所示。



图 6-108　MySQL 数据库连接的配置

在图 6-107 的 SQL 框中编写获取数据表 temporary 中字段 id 和 salesArea 数据的 SQL 语句，然后单击“预览”按钮，查看数据表 temporary 中的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-109 和图 6-110 所示。



图 6-109　编写 SQL 语句

从图 6-110 中可以看出，数据表 temporary 中的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

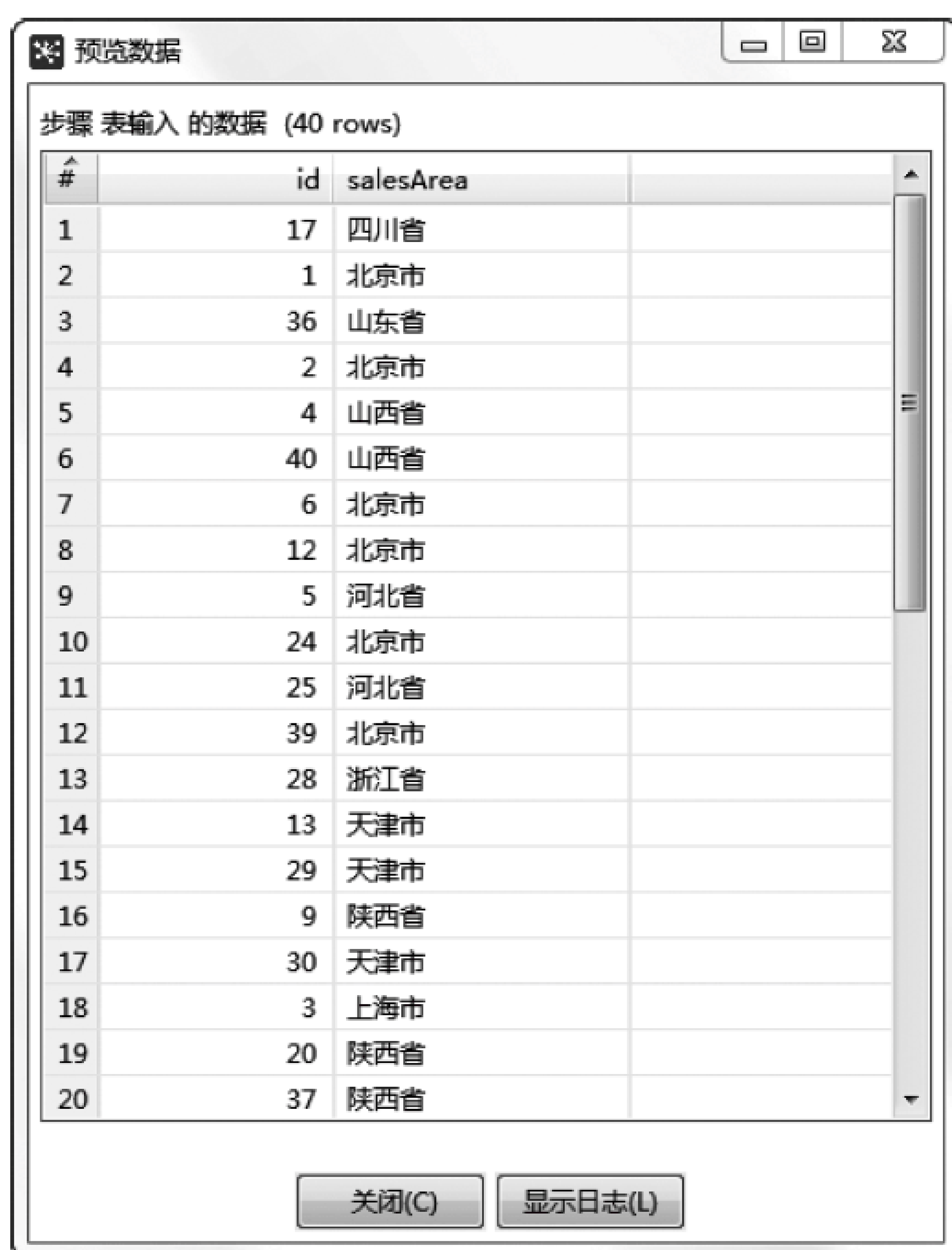


图 6-110 预览数据

20. 配置“字段选择”控件

双击图 6-106 中的“字段选择”控件,进入“选择/改名值”界面,如图 6-111 所示。



图 6-111 “选择/改名值”界面

在图 6-111 中的“选择和修改”选项卡的“字段”处手动添加所需字段,这里添加字段 id 和 salesArea,用于与数据表 company 中的数据进行合并。“选择和修改”选项卡的配置如图 6-112 所示。

在图 6-112 中单击“确定”按钮,完成“字段选择”控件的配置。

21. 配置“排序记录”控件

双击图 6-106 中的“排序记录”控件,进入“排序记录”界面,如图 6-113 所示。



图 6-112　“选择和修改”选项卡的配置



图 6-113　“排序记录”界面

在图 6-113 的“字段”框中添加字段 id，以此字段为基础对所有数据进行升序排序，具体如图 6-114 所示。



图 6-114　配置“排序记录”控件

在图 6-114 中单击“确定”按钮，完成“排序记录”控件的配置。

22. 配置“表输入 2”控件

双击图 6-106 中的“表输入 2”控件，进入“表输入”界面，如图 6-115 所示。



图 6-115 “表输入”界面

在图 6-115 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-116 所示。



图 6-116 MySQL 数据库连接的配置

在图 6-115 的 SQL 框中编写 SQL 语句，用于查询数据表 company 中的全部数据，然后单击“预览”按钮，查看数据表 company 中的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-117 和图 6-118 所示。



图 6-117　编写 SQL 语句


预览数据

步骤 表输入 2 的数据 (40 rows)

#	id	salesArea	brand	model	unitPrice	number
1	1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	7	石家庄市	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	10	大连市	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	11	秦皇岛市	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	14	青岛市	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	15	西安市	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	18	大同市	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39

关闭(C)　显示日志(L)

图 6-118　预览数据

从图 6-118 中可以看出，数据表 company 中的数据已成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入 2”控件的配置。

23. 配置“字段选择 2”控件

双击图 6-106 中的“字段选择 2”控件，进入“选择/改名值”界面，如图 6-119 所示。

在图 6-119 中的“选择和修改”选项卡的“字段”处手动添加所需字段，这里添加字段 id、brand、model、unitPrice、number，之后通过字段 id 将数据表 temporary 中的字段 salesArea



图 6-119 “选择/改名值”界面

和数据表 company 中的数据进行合并。“选择和修改”选项卡的配置如图 6-120 所示。



图 6-120 “选择和修改”选项卡的配置

在图 6-120 中单击“确定”按钮，完成“字段选择 2”控件的配置。

24. 配置“排序记录 2”控件

双击图 6-106 中的“排序记录 2”控件，进入“排序记录”界面，具体如图 6-121 所示。



图 6-121 “排序记录”界面

在图 6-121 的“字段”框中添加字段 id，以此字段为基础对全部数据进行升序排序，具体如图 6-122 所示。



图 6-122　配置“排序记录 2”控件

在图 6-122 中单击“确定”按钮，完成“排序记录 2”控件的配置。

25. 配置“记录集连接”控件

双击图 6-106 中的“记录集连接”控件，进入“合并排序”界面，如图 6-123 所示。



图 6-123　“合并排序”界面

在图 6-123 中“第一个步骤：”后的下拉列表中选择“排序记录”，在“第二个步骤：”后的下拉列表中选择“排序记录 2”；在“第一个步骤的连接字段”和“第二个步骤的连接字段”处添加连接字段，这里添加的连接字段是 id 和 id，用于将“排序记录”控件中的数据与“排序记录 2”控件中的数据合并连接；在“连接类型”处的下拉列表中选择连接类型，这里选择的是 FULL OUTER，即完全外连接，具体如图 6-124 所示。

在图 6-124 中单击“确定”按钮，完成“记录集连接”控件的配置。

26. 配置“字段选择 3”控件

双击图 6-106 中的“字段选择 3”控件，进入“选择/改名值”界面，如图 6-125 所示。



图 6-124 配置“记录集连接”控件



图 6-125 “选择/改名值”界面

在图 6-125 中“选择和修改”选项卡的“字段”处手动添加所需字段，由于源数据表 company 中包含字段 id、salesArea、brand、model、unitPrice、number，因此需要获取“记录集连接”控件流中的字段 id、salesArea、brand、model、unitPrice、number，用于与源数据表 company 中的字段对应，具体如图 6-126 所示。



图 6-126 “字段选择 3”控件的配置

在图 6-126 中单击“确定”按钮，完成“字段选择 3”控件的配置。

27．配置“表输出”控件

双击图 6-106 中的“表输出”控件，进入“表输出”界面，具体如图 6-127 所示。



图 6-127　“表输出”界面

在图 6-127 中单击“新建”按钮，配置数据库连接(所连接的数据库 transform 须提前创建，这里不赘述)，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-128 所示。



图 6-128　MySQL 数据库连接的配置

单击图 6-127 中目标表右侧的“浏览”按钮，指定输出目标表，即数据表 company_new(该表须提前创建，这里不演示)；勾选“指定数据库字段”复选框，用于将数据表 temporary 的字段与“字段选择 3”控件输出流中的字段进行匹配，如图 6-129 所示。



图 6-129　指定输出目标表和勾选“指定数据库字段”复选框

单击图 6-129 中的“数据库字段”选项卡，具体如图 6-130 所示。



图 6-130　“数据库字段”选项卡

在图 6-130 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，具体如图 6-131 所示。



图 6-131　“映射匹配”对话框

在图 6-131 中依次选中“源字段”中的字段和“目标字段”中对应的字段，然后单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 6-132 所示。

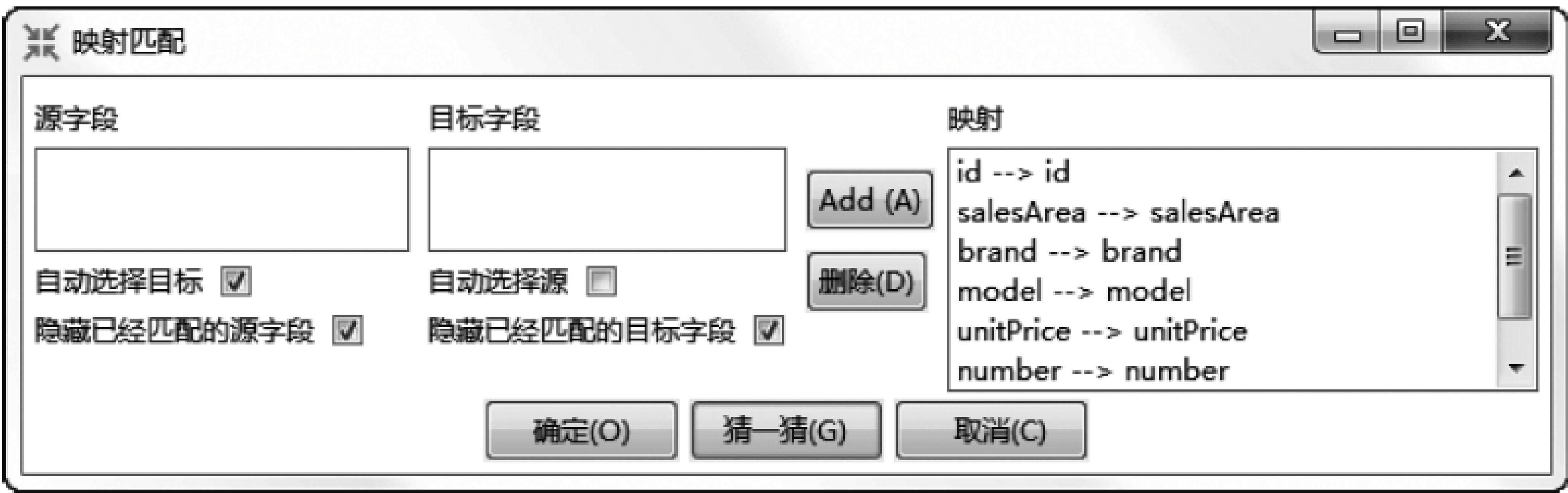


图 6-132　设置映射匹配

在图 6-132 中单击“确定”按钮，完成“源字段”与“目标字段”的映射匹配。“表输出”控件配置的效果图如图 6-133 所示。



图 6-133　“表输出”控件配置的效果图

在图 6-133 中单击“确定”按钮，完成“表输出”控件的配置。

28. 打开 Kettle 工具，创建作业

使用 Kettle 工具创建作业 granularity，并添加 Start 控件、“转换”控件、“成功”控件以及作业跳连接线，具体效果如图 6-134 所示。

29. 配置 Start 控件

双击图 6-134 中的 Start 控件，进入“作业定时调度”界面，具体如图 6-135 所示。

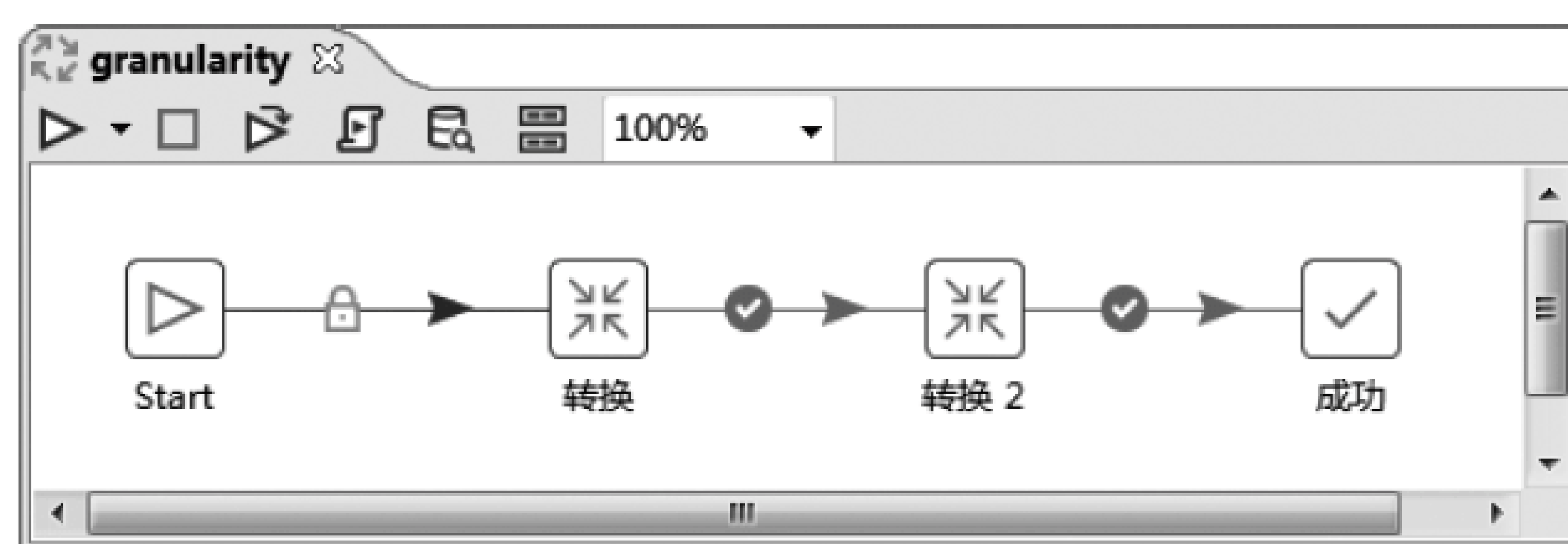


图 6-134　创建作业 granularity



图 6-135　“作业定时调度”界面

在图 6-135 中可以设置时间调度以执行作业，这里使用默认配置，不作修改。单击“确定”按钮，完成 Start 控件的配置。

30. 配置“转换”控件

双击图 6-134 中的“转换”控件，进入“转换”界面，具体如图 6-136 所示。



图 6-136　“转换”界面

在图 6-136 中单击“浏览”按钮,选择添加转换 granularity 至作业中,如图 6-137 所示。

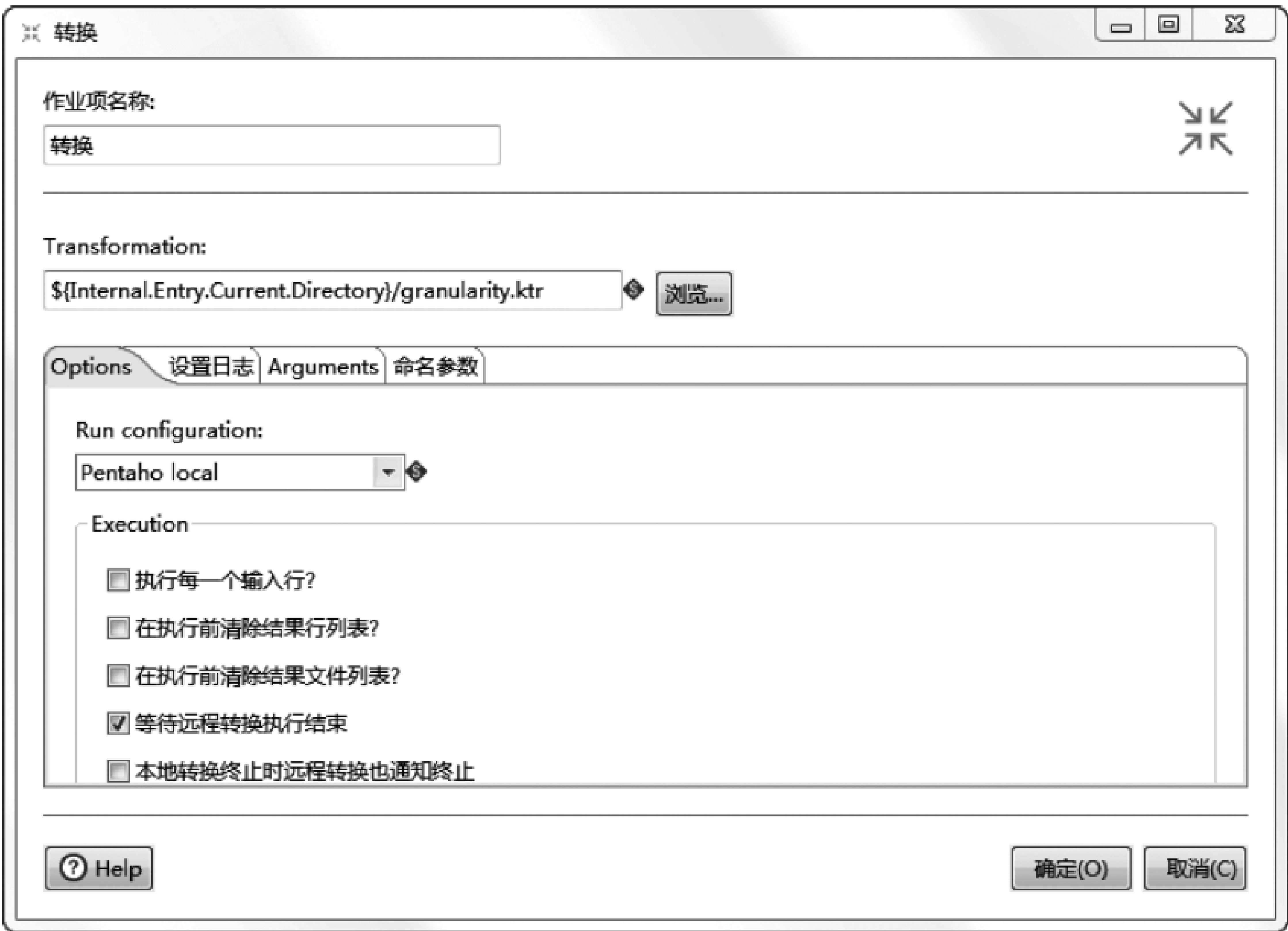


图 6-137 添加转换 granularity 至作业中

在图 6-137 中单击“确定”按钮,完成“转换”控件的配置。

31. 配置“转换 2”控件

双击图 6-134 中的“转换 2”控件,进入“转换”界面,具体如图 6-138 所示。



图 6-138 “转换”界面

在图 6-138 中单击“浏览”按钮，选择添加转换 granularity_merge 至作业中，如图 6-139 所示。

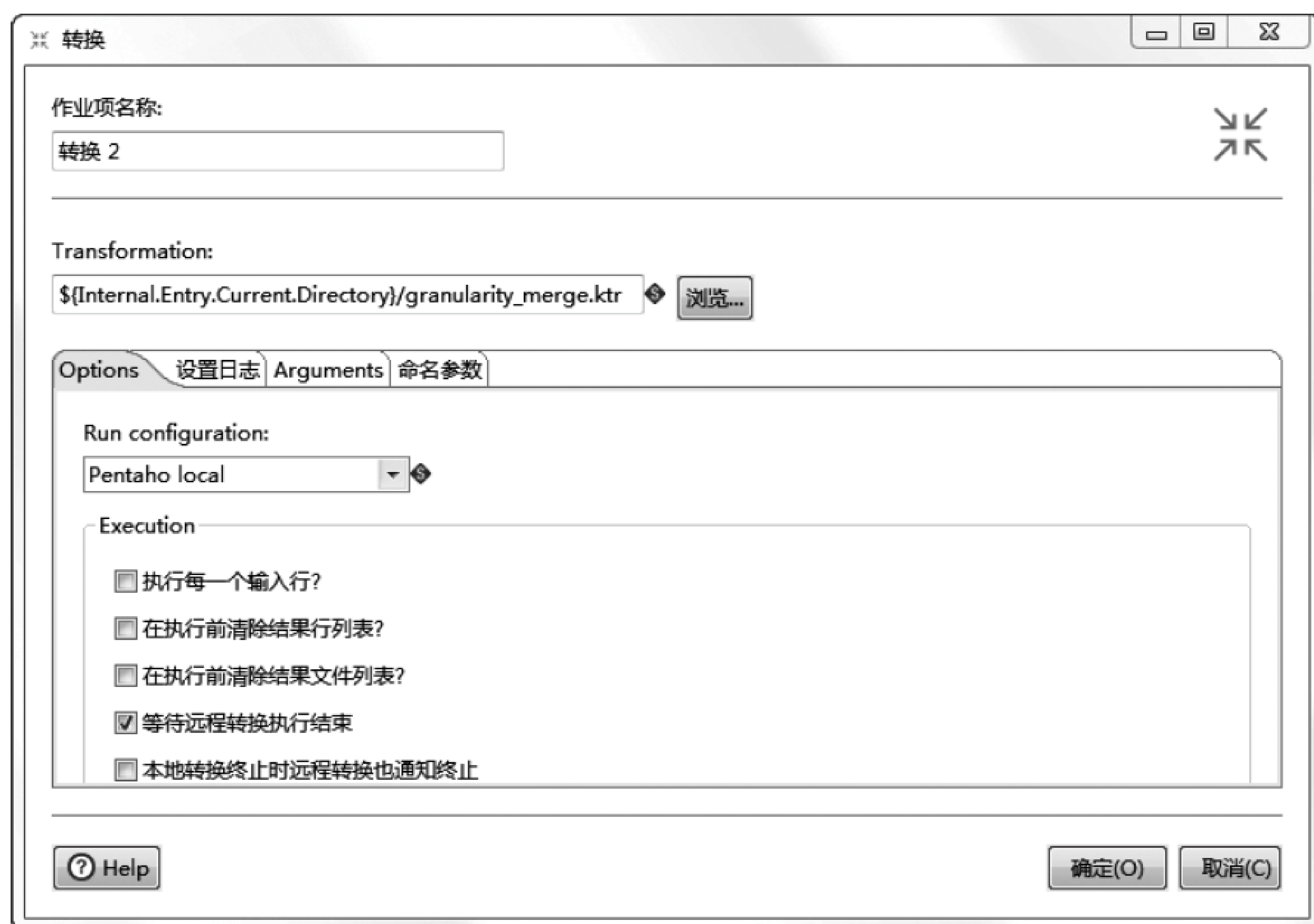


图 6-139 添加转换 granularity_merge 至作业中

在图 6-139 中单击“确定”按钮，完成“转换 2”控件的配置。

32．运行作业 granularity

单击作业工作区顶部的▷按钮，运行创建的作业 granularity，实现将数据表 company 中字段为 salesArea 的数据都统一成省份级，并存储到数据表 company_new 中，具体如图 6-140 所示。

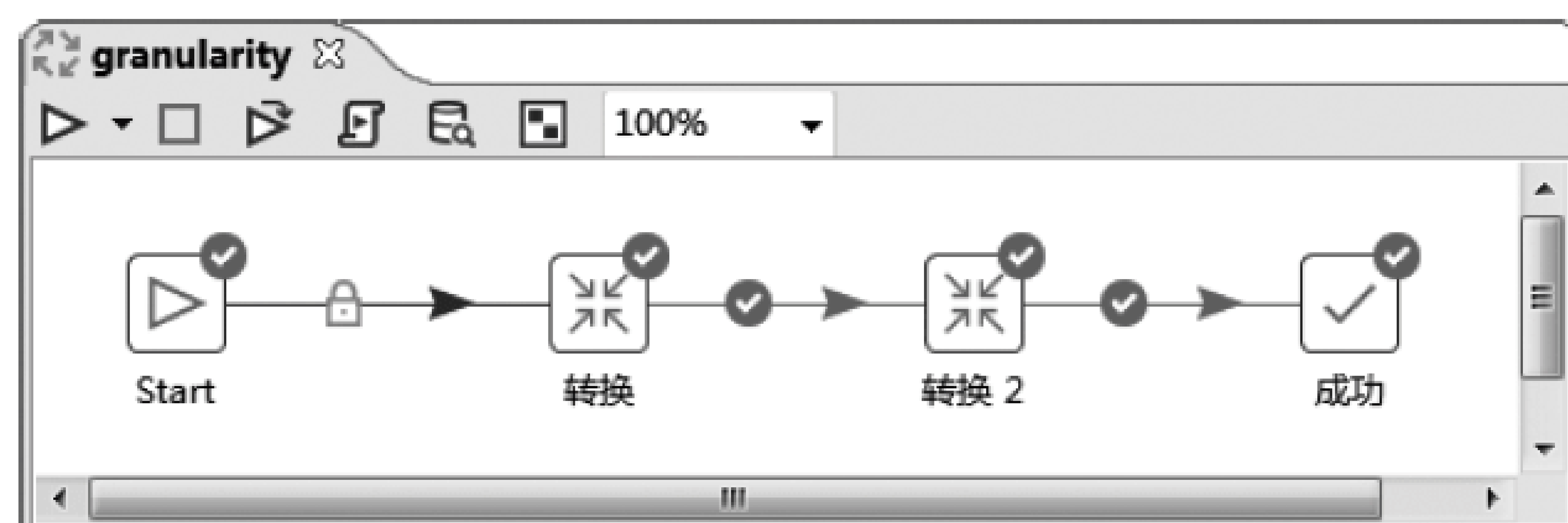


图 6-140 运行作业 granularity

从图 6-140 的对勾图标可以看出，作业 granularity 运行成功。

33．查看数据表 company_new 中的数据

通过 SQLyog 工具，查看数据表 company_new 中字段为 salesArea 的数据是否都为省份级，查看结果如图 6-141 所示(只展示部分数据)。

从图 6-141 中可以看出，数据表 company_new 中字段为 salesArea 的数据都为省份级。

id	salesArea	brand	model	unitPrice	number
1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	辽宁省	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	河北省	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	山东省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	陕西省	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	山西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39
21	上海市	华为	华为Mate 20 X（8GB/256GB/全网通/5G版）	6199	42
22	黑龙江省	OPPO	OPPO R17（8GB RAM/全网通）	2499	29
23	广东省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	36
24	北京市	三星	三星Galaxy S10+（8GB RAM/陶瓷版/全网通）	7499	27
25	河北省	苹果	苹果iPhone 11（4GB/128GB/全网通）	5999	20

图 6-141　数据表 company_new

6.4　数据的商务规则计算

不同的企业有不同的业务规则和数据指标，这些指标应该计算完存储到数据仓库中，供企业决策者进行分析，从而得出战略性的企业决策。例如，A 公司和 B 公司的总公司想得知各省份的手机日销售额，这就属于一个商务规则。可以通过对数据表 company_new 中的数据进行处理和计算，得出总公司需要的各省份的手机日销售额。数据表 company_new 中的数据内容如图 6-142 所示。

下面通过 Kettle 工具对数据表 company_new 中的数据进行商务规则的计算，即对数据表 company_new 中的数据进行相关处理和计算，从而得出手机在各省份的日销售额，并存储于数据表 regional_sales 中，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 total，并添加“表输入”控件、“字段选择”控件、“计算器”控件、“排序记录”控件、“分组”控件、“唯一行(哈希值)”控件、“表输出”控件以及 Hop 跳连接线，具体效果如图 6-143 所示。

2. 配置“表输入”控件

双击图 6-143 中的“表输入”控件，进入“表输入”界面，如图 6-144 所示。

在图 6-144 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-145 所示。

id	salesArea	brand	model	unitPrice	number
1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	辽宁省	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	河北省	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	山东省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	陕西省	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	山西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39
21	上海市	华为	华为Mate 20 X（8GB/256GB/全网通/5G版）	6199	42
22	黑龙江省	OPPO	OPPO R17（8GB RAM/全网通）	2499	29
23	广东省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	36
24	北京市	三星	三星Galaxy S10+（8GB RAM/陶瓷版/全网通）	7499	27
25	河北省	苹果	苹果iPhone 11（4GB/128GB/全网通）	5999	20

图 6-142　数据表 company_new 中的数据内容

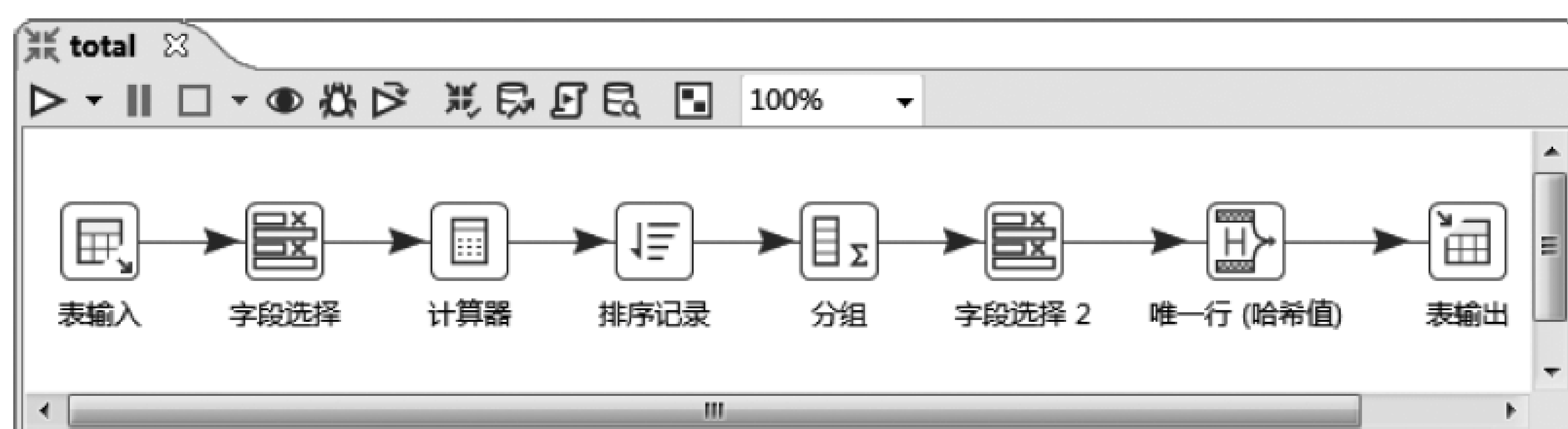


图 6-143　创建转换 total

表输入

步骤名称 表输入

数据库连接 编辑... 新建... Wizard...

SQL 获取SQL查询语句...

SELECT <values> FROM <table name> WHERE <conditions>

行1 列0

允许简易转换

替换 SQL 语句里的变量

从步骤插入数据

执行每一行?

记录数量限制 0

Help 确定(O) 预览(P) 取消(C)

图 6-144　“表输入”界面



图 6-145　MySQL 数据库连接的配置

在图 6-144 的 SQL 框中编写查询数据表 company_new 中全部数据的 SQL 语句，然后单击“预览”按钮，查看数据表 company_new 中的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 6-146 和图 6-147 所示。



图 6-146　编写 SQL 语句

从图 6-147 中可以看出，数据表 company_new 中的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“字段选择”控件

双击图 6-143 中的“字段选择”控件，进入“选择/改名值”界面，如图 6-148 所示。

在图 6-148 中的“选择和修改”选项卡的“字段”处手动添加所需字段，这里添加字段 id、

预览数据

步骤 表输入 的数据 (40 rows)

#	id	salesArea	brand	model	unitPrice	number
1	1	北京市	华为	华为nova 5 Pro（8GB/128GB/全网通）	2999	31
2	2	北京市	苹果	苹果iPhone 11 Pro（6GB/64GB/全网通）	8699	50
3	3	上海市	苹果	苹果iPhone 11 Pro（6GB/256GB/全网通）	9999	20
4	4	山西省	苹果	苹果iPhone 11（4GB/64GB/全网通）	5499	20
5	5	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	27
6	6	北京市	三星	三星GALAXY Note 10+（12GB/256GB/全网通/5G版）	7999	32
7	7	河北省	OPPO	OPPO Reno2（8GB/128GB/全网通）	2999	20
8	8	黑龙江省	华为	华为Mate30（8GB/128GB/全网通/5G版/玻璃版）	4999	34
9	9	陕西省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	38
10	10	辽宁省	OPPO	OPPO Reno 10倍变焦版（8GB/256GB/全网通）	4299	35
11	11	河北省	vivo	vivo NEX 3（8GB/256GB/全网通/5G版）	5698	26
12	12	北京市	vivo	vivo NEX 3（8GB/128GB/全网通）	4998	26
13	13	天津市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	18
14	14	山东省	vivo	vivo iQOO Pro（8GB/128GB/5G全网通）	3798	25
15	15	陕西省	华为	华为P30 Pro（8GB/128GB/全网通）	4988	30
16	16	重庆市	小米	小米9（8GB/256GB/全网通）	2999	26
17	17	四川省	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	35
18	18	山西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	22
19	19	重庆市	苹果	苹果iPhone 11 Pro Max（6GB/64GB/全网通）	9599	38
20	20	陕西省	小米	小米9 Pro（8GB/256GB/全网通/5G版）	3799	39

关闭(C) 显示日志(L)

图 6-147 预览数据

选择/改名值

步骤名称 字段选择

选择和修改 移除 元数据

字段：

#	字段名称	改名成	长度	精度
1				

获取选择的字段 列映射

包含未指定的列,按名称排序

Help 确定(O) 取消(C)

图 6-148 “选择/改名值”界面

salesArea、unitPrice 和 number，用于后续的计算处理，具体配置如图 6-149 所示。

在图 6-149 中选择“元数据”选项卡，切换到“元数据”选项卡界面，如图 6-150 所示。

在图 6-150 中添加需要改变元数据的字段，由于数据表 company_new 中字段 unitPrice 的数据类型为 varchar(字符)，字段 number 的数据类型为 int，又由于字段类型不同的数据无法进行计算，因此需要将字段 unitPrice 的数据类型改为 Integer，从而进行后续的计算操作。“元数据”选项卡的配置如图 6-151 所示。

在图 6-151 中单击“确定”按钮，完成“字段选择”控件的配置。

4. 配置“计算器”控件

双击图 6-143 中的“计算器”控件，进入“计算器”界面，如图 6-152 所示。



图 6-149　添加所需字段



图 6-150　“元数据”选项卡界面



图 6-151　“元数据”选项卡的配置

在图 6-152 中的“字段”处添加一个新字段 salesAmount，用于存储计算出的手机日销售额；在“字段 A”和“字段 B”处的下拉选项中分别选择 unitPrice（销售价格）和 number（销售数量）字段；在“计算”处的下拉选项中选择“A * B”，即表示将字段 A 与字段 B 进行相乘计算，具体如图 6-153 所示。

在图 6-153 中单击“确定”按钮，完成“计算器”控件的配置。



图 6-152　“计算器”界面

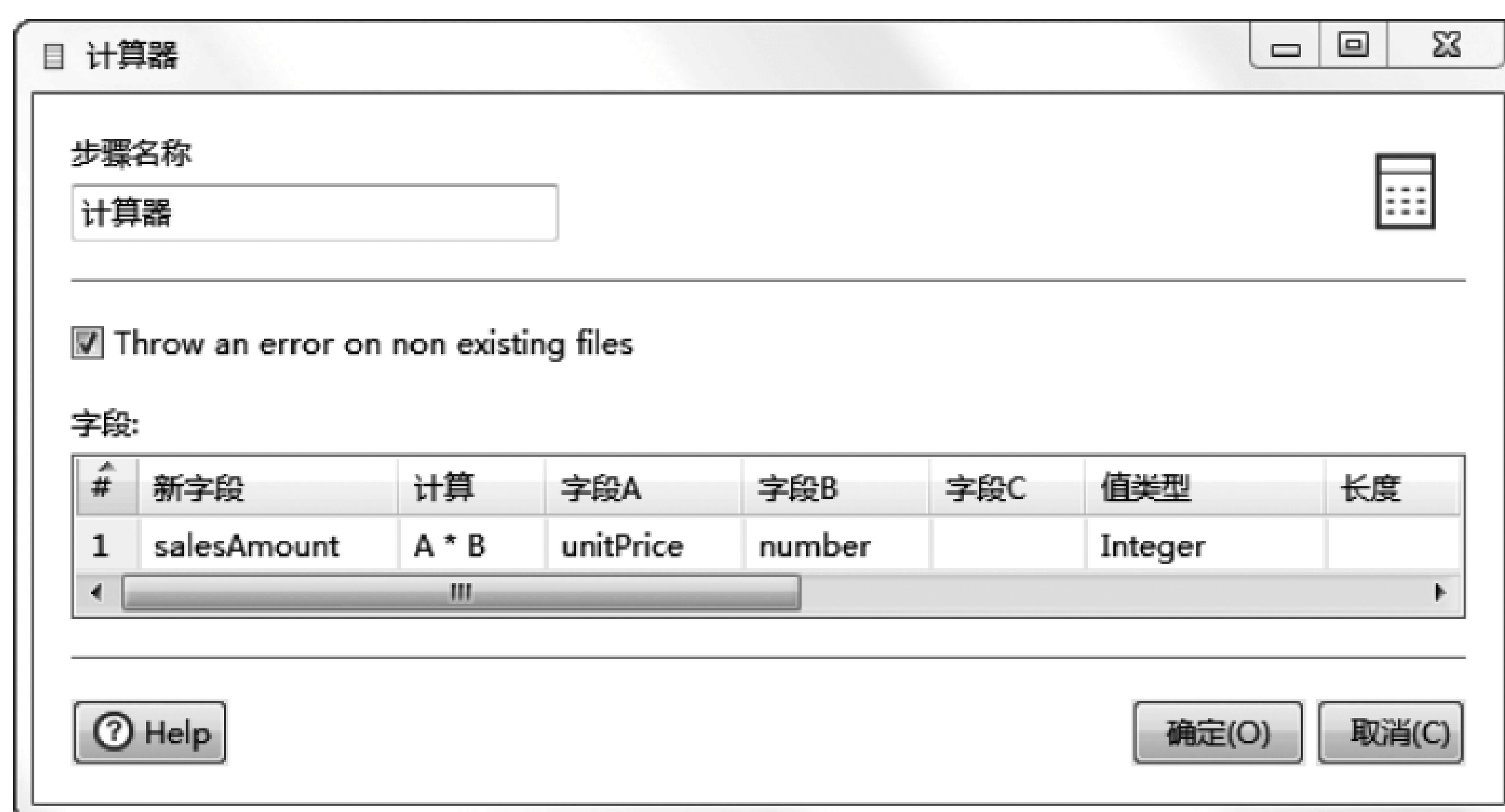


图 6-153　“计算器”控件配置的效果图

5. 配置“排序记录”控件

双击图 6-143 中的“排序记录”控件，进入“排序记录”界面，如图 6-154 所示。



图 6-154　“排序记录”界面

在图 6-154 中的“字段”框中添加字段 salesArea，以该字段为基础对所有数据进行升序排序，具体如图 6-155 所示。



图 6-155　配置“排序记录”控件

在图 6-155 中单击“确定”按钮，完成“排序记录”控件的配置。

6. 配置“分组”控件

双击图 6-143 中的“分组”控件，进入“分组”界面，如图 6-156 所示。



图 6-156　“分组”界面

在图 6-156 中的“构成分组的字段”处添加分组字段 salesArea，将字段 salesArea 相同的数据分为一组，便于进行“聚合”操作；在“聚合”处添加一个新字段 total，用于存储各省份的手机销售总额，具体如图 6-157 所示。

在图 6-157 中单击“确定”按钮，完成“分组”控件的配置。



图 6-157 配置“分组”控件

7. 配置“字段选择 2”控件

双击图 6-143 中的“字段选择 2”控件,进入“选择/改名值”界面,如图 6-158 所示。



图 6-158 “选择/改名值”界面

在图 6-158 的“选择和修改”选项卡中选择需要的字段,这里选择的字段是 salesArea 和 total,用于后续在“表输出”控件中进行输出操作,具体如图 6-159 所示。



图 6-159 “字段选择 2”控件的配置

在图 6-159 中单击“确定”按钮，完成“字段选择 2”控件的配置。

8. 配置“唯一行”控件

双击图 6-143 中的“唯一行”控件，进入“唯一行”界面，如图 6-160 所示。



图 6-160　“唯一行”界面

在图 6-160 中的“用来比较的字段”处添加要去重的字段，因为“字段选择 2”控件流中字段 total 的数据有重复，所以需要进行去重操作，这里选择去重的字段为 total，具体如图 6-161 所示。

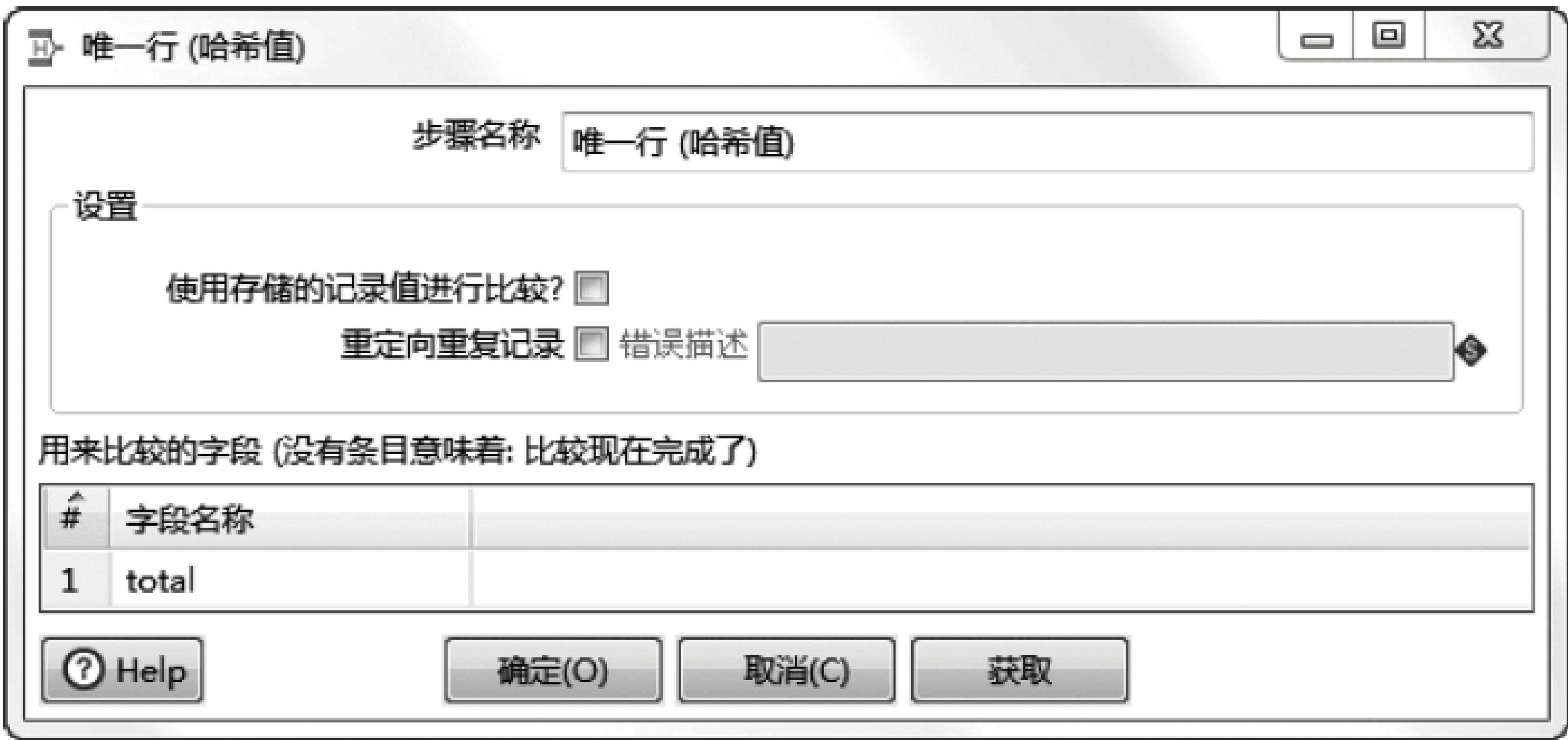


图 6-161　“唯一行”控件的配置

在图 6-161 中单击“确定”按钮，完成“唯一行”控件的配置。

9. 配置“表输出”控件

双击图 6-143 中的“表输出”控件，进入“表输出”界面，具体如图 6-162 所示。

在图 6-162 中单击“新建”按钮，配置数据库连接(所连接的数据库 transform 须提前创建，这里不赘述)，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 6-163 所示。

单击图 6-162 中目标表右侧的“浏览”按钮，指定输出目标表，即数据表 regional_sales (该表须提前创建，这里不演示)；勾选“指定数据库字段”复选框，用于将数据表 regional_sales 的字段与“唯一行”控件输出流中的字段进行匹配，如图 6-164 所示。

单击图 6-164 中的“数据库字段”选项卡，具体如图 6-165 所示。



图 6-162 “表输出”界面



图 6-163 MySQL 数据库连接的配置

在图 6-165 中单击“输入字段映射”按钮，弹出“映射匹配”对话框，具体如图 6-166 所示。

在图 6-166 中依次选中“源字段”中的字段和“目标字段”中对应的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，用于将数据表 company_new 中的字段 salesArea、total 与目标数据表中的字段 salesArea、total 进行匹配，具体如图 6-167 所示。

在图 6-167 中单击“确定”按钮，完成“源字段”与“目标字段”的映射匹配。“表输出”控件配置的效果图如图 6-168 所示。

在图 6-168 中单击“确定”按钮，完成“表输出”控件的配置。



图 6-164　指定输出目标表和勾选“指定数据库字段”复选框



图 6-165　“数据库字段”选项卡



图 6-166　“映射匹配”对话框



图 6-167 设置映射匹配



图 6-168 “表输出”控件配置的效果图

10. 运行转换 total

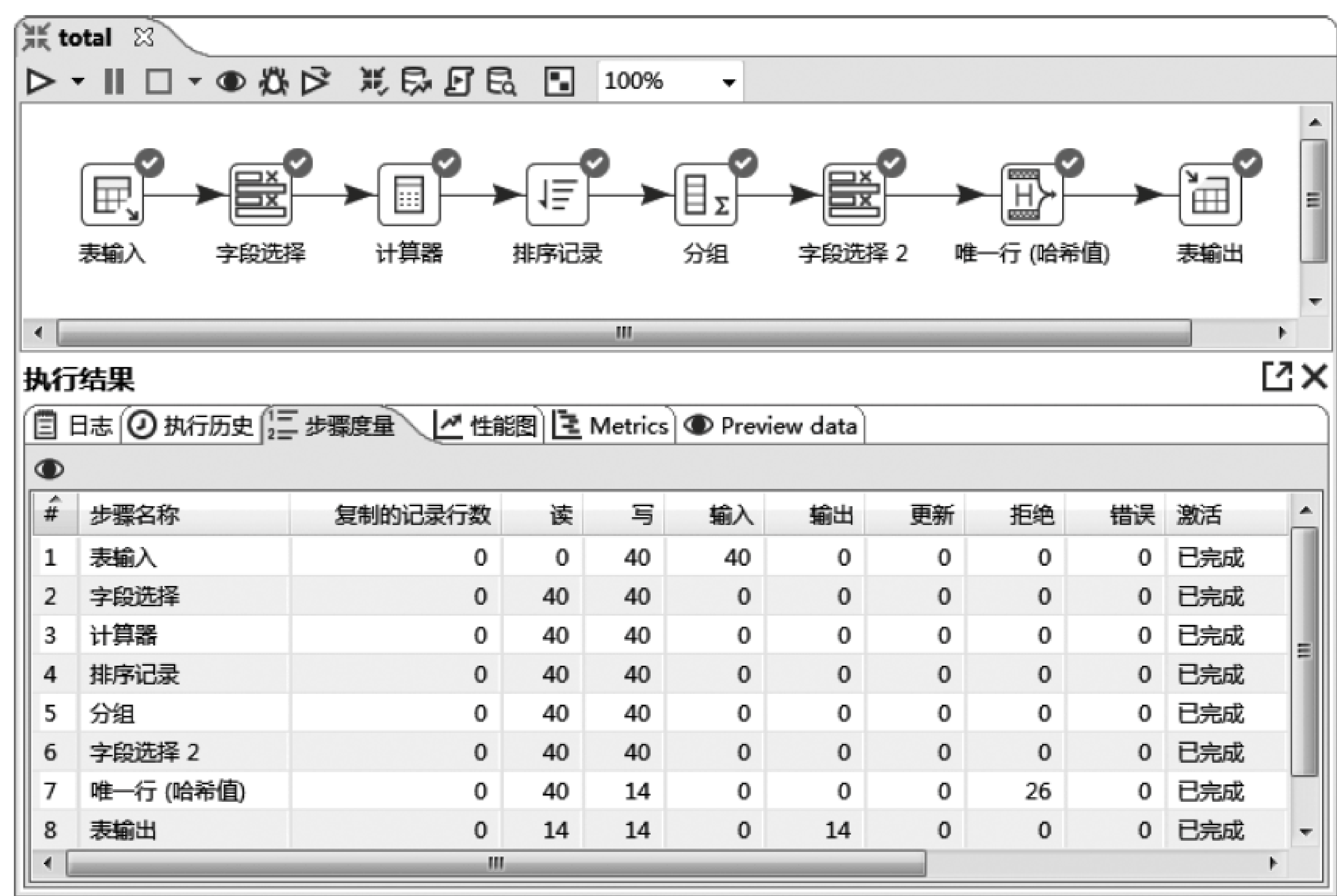
单击转换工作区顶部的▷按钮，运行创建的转换 total，实现对数据表 company_new 中数据进行商务规则的计算，从而得出手机在各省的日销售额，并存储于数据表 regional_sales 中，具体如图 6-169 所示。

从图 6-169 中执行结果的“步骤度量”可以看出，“表输入”控件输入 40 条数据并写入该控件；“字段选择”控件从“表输入”控件中读取 40 条数据并写入该控件；“计算器”控件从“字段选择”控件中读取 40 条数据并写入该控件；“排序记录”控件从“计算器”控件中读取 40 条数据并写入该控件；“分组”控件从“排序记录”控件中读取 40 条数据并写入该控件；“字段选择 2”控件从“分组”控件中读取 40 条数据并写入该控件；“唯一行(哈希值)”控件从“字段选择 2”控件中读取 40 条数据，并将符合要求的 14 条数据写入该控件；“表输出”控件从“唯一行(哈希值)”控件中读取 14 条数据并写入该控件，最终进行输出。

11. 查看数据表 regional_sales 中的数据

通过 SQLyog 工具，查看数据表 regional_sales 中是否已存储各省份的手机日销售额数据，查看结果如图 6-170 所示。

从图 6-170 中可以看出，数据表 regional_sales 中已经存储了各省份的手机日销售额数据。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	表输入	0	0	40	40	0	0	0	0	已完成
2	字段选择	0	40	40	0	0	0	0	0	已完成
3	计算器	0	40	40	0	0	0	0	0	已完成
4	排序记录	0	40	40	0	0	0	0	0	已完成
5	分组	0	40	40	0	0	0	0	0	已完成
6	字段选择 2	0	40	40	0	0	0	0	0	已完成
7	唯一行 (哈希值)	0	40	14	0	0	0	26	0	已完成
8	表输出	0	14	14	0	14	0	0	0	已完成

图 6-169　运行转换 total

salesArea	total
上海市	820266
北京市	1357773
四川省	335965
天津市	457524
山东省	210930
山西省	451250
广东省	345564
河北省	527815
浙江省	197171
湖北省	110963
辽宁省	307047
重庆市	651698
陕西省	667977
黑龙江省	242437

图 6-170　数据表 regional_sales

6.5　本章小结

本章主要讲解数据转换的相关知识，包括多数据源合并、不一致数据转换、数据颗粒度的转换以及数据的商务规则计算。希望读者通过本章的学习，可以掌握数据的转换操作，实现将企业中的数据进行规范化处理。

6.6　本章习题

一、填空题

1. ________是数据清洗过程的重要步骤之一。

2. ________主要是将不同业务系统中的相同类型的数据进行统一。

3. 一般情况下会将业务系统数据按数据仓库粒度进行聚合,这个过程被称为________。

二、操作题

现有一个文本文件 personnel_data.txt,包含字段 id、name、id_number、household_register 和 salary,具体内容如下所示。

id	name	id_number	household_register	salary
1	zhangsan	110101199003074135	北京市	600
2	bob	110102200010250451	北京市	500
3	wangwu	120101199308198578	天津市	500
4	zhaoliu	130102198809275932	石家庄市	200
5	sunqi	310101199001207798	上海市	800
6	rose	330102198610203149	杭州市	500
7	wujiu	500101199210039242	重庆市	300
8	zhengshi	500101199110038661	重庆市	200
9	allen	140202199208066697	大同市	300
10	lisi	130201199603157822	唐山市	200
11	mary	370634198707208961	烟台市	200
12	daniel	430103198405258300	长沙市	300
13	lily	110105198501308898	北京市	600
14	zhouba	210302198410189892	鞍山市	200
15	william	310104199307038877	上海市	500
16	amy	320102199412209710	南京市	200

通过使用 Kettle 工具,实现以下功能:

(1) 对文本文件 personnel_data.txt 中的数据进行数据粒度的转换,即将文本文件 personnel_data.txt 中字段为 household_register 的数据统一成省份,并输出到文本文件 personnel_data_new.txt 中。

(2) 对文本文件 personnel_data_new.txt 中字段为 salary 的数据进行商务规则计算,即计算每个人的月薪(以 22 天工作日计算),最终输出到文本文件 personnel_data_monthly_salary.txt 中。

第 7 章 数 据 加 载

学习目标

（1）熟悉数据的增量加载

（2）掌握数据的全量加载

（3）理解数据的批处理

数据的预处理过程，除包括数据抽取、数据本身的清洗与检验以及数据转换操作外，还包括数据加载操作。数据加载是数据预处理过程的最后一个步骤，主要负责将清洗检验、转换后的高质量数据加载到目标数据库中。本章将针对数据加载的相关知识进行详细讲解。

7.1 数据的加载机制

数据的加载机制与数据的抽取机制类似。数据的加载机制可以分为全量加载和增量加载。其中，全量加载是指将目标数据表中的数据全部删除后进行数据加载的操作；而增量加载是指目标表只加载源数据表中变化的数据，包含新增、修改和删除的数据。本节将对数据的全量加载和增量加载进行讲解。

7.1.1 全量加载

从技术角度来说，全量加载比增量加载的操作要简单很多，即只需要在数据加载之前将目标数据表进行清空，再将源数据表中的数据全部加载到目标表中。

假设有两张数据表，分别为 full_source 和 full_target，其中 full_source 为源数据表，full_target 为目标数据表。full_source 和 full_target 的具体内容分别如图 7-1 和图 7-2 所示。

no	book_name	author	tag
1	小王子	圣艾克苏佩里	外国文学
2	解忧杂货店	东野圭吾	小说
3	白夜行	东野圭吾	小说
4	梦里花落知多少	郭敬明	小说
5	看见	柴静	随笔
6	围城	钱钟书	小说

图 7-1 数据表 full_source

下面通过 Kettle 工具将数据表 full_source 中的数据全量加载到数据表 full_target 中，

no	book_name	author	tag
1	小王子	圣艾克苏佩里	外国文学
2	解忧杂货店	东野圭吾	小说
3	看见	柴静	随笔

图 7-2 数据表 full_target

具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 full_load，并添加“执行 SQL 脚本”控件、“表输入”控件、“表输出”控件以及 Hop 跳连接线，具体效果如图 7-3 所示。

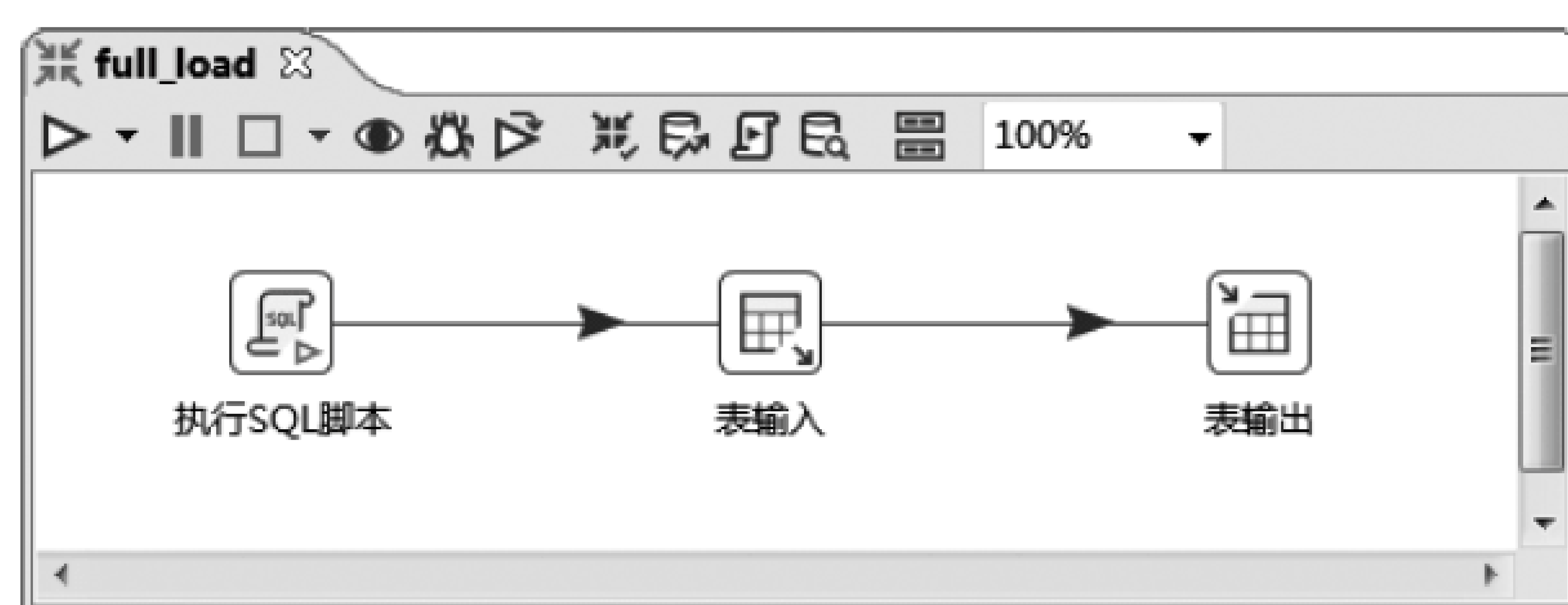


图 7-3 创建转换 full_load

2. 配置“执行 SQL 脚本”控件

双击图 7-3 中的“执行 SQL 脚本”控件，进入“执行 SQL 语句”界面，如图 7-4 所示。



图 7-4 “执行 SQL 语句”界面

在图 7-4 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 7-5 所示。



图 7-5　MySQL 数据库连接的配置

在图 7-4 的 SQL 框中编写删除数据表 full_target 中数据的 SQL 语句，具体如图 7-6 所示。



图 7-6　编写 SQL 语句

在图 7-6 中单击“确定”按钮，完成“执行 SQL 脚本”控件的配置。

3. 配置“表输入”控件

双击图 7-3 中的“表输入”控件，进入“表输入”界面，如图 7-7 所示。



图 7-7 “表输入”界面

在图 7-7 的 SQL 框中编写查询数据表 full_source 数据的 SQL 语句，然后单击“预览”按钮，查看数据表 full_source 的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 7-8 和图 7-9 所示。



图 7-8 编写 SQL 语句

从图 7-9 中可以看出，full_source 的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

4. 配置“表输出”控件

双击图 7-3 中的“表输出”控件，进入“表输出”界面，如图 7-10 所示。

单击图 7-10 中目标表处的“浏览”按钮，选择输出的目标表，即数据表 full_target，这里不需要新建数据库连接，在数据库连接后的下拉列表中选择已创建的数据库连接即可，具体如图 7-11 所示。



图 7-9　预览数据



图 7-10　“表输出”界面

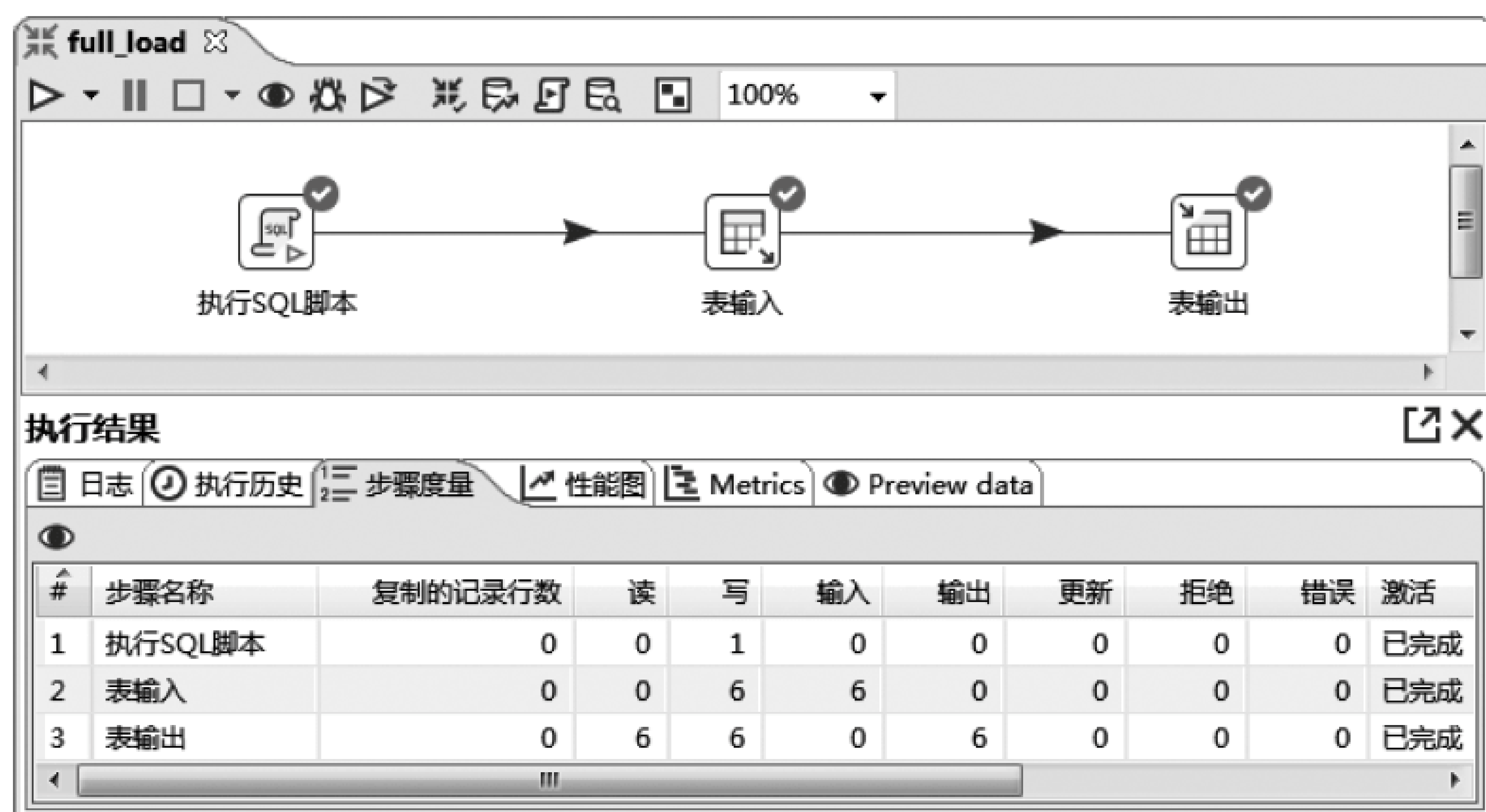


图 7-11　选择目标表 full_target

在图 7-11 中单击“确定”按钮，完成“表输出”控件的配置。

5. 运行转换 full_load

单击转换工作区顶部的▷按钮，运行创建的转换 full_load，实现将数据表 full_source 中的数据全量加载到数据表 full_target 中，具体如图 7-12 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	执行SQL脚本	0	0	1	0	0	0	0	0	已完成
2	表输入	0	0	6	6	0	0	0	0	已完成
3	表输出	0	6	6	0	6	0	0	0	已完成

图 7-12　运行转换 full_load

从图 7-12 中执行结果的“步骤度量”可以看出，“执行 SQL 脚本”控件写入 1 条 SQL 语句；“表输入”控件输入 6 条数据，并写入该控件；“表输出”控件从“表输入”控件中读取 6 条数据，并写入该控件，最终进行输出。

6. 查看数据表 full_target 中的数据

通过 SQLyog 工具，查看数据表 full_target 是否已成功加载到数据，查看结果如图 7-13 所示。

no	book_name	author	tag
1	小王子	圣艾克苏佩里	外国文学
2	解忧杂货店	东野圭吾	小说
3	白夜行	东野圭吾	小说
4	梦里花落知多少	郭敬明	小说
5	看见	柴静	随笔
6	围城	钱钟书	小说

图 7-13　数据表 full_target

从图 7-13 中可以看出，数据表 full_target 中已经加载到数据，说明我们成功实现了将数据表 full_source 中的数据全量加载到数据表 full_target 中。

7.1.2　增量加载

增量加载是指目标表仅加载源数据表中新增和发生变化的数据。优秀的增量加载机制不但能够将业务系统中的变化数据按一定的频率准确地捕获并加载到目标表中，同时还不会对业务系统造成太大的压力，也不会影响现有业务。

假设有两张数据表，分别为 incremental_source 和 incremental_target，其中 incremental_source 为源数据表；incremental_target 为目标数据表。数据表 incremental_source 和 incremental_target 的表结构、数据都是相同的，具体如图 7-14 和图 7-15 所示。

id	name	age	create_time
1	Isabella	18	2019-08-20 13:14:20
2	Jack	20	2019-08-20 13:14:21
3	Nicholas	22	2019-08-20 13:14:22
4	Jasmine	19	2019-08-20 13:14:23
5	Mia	20	2019-08-20 13:14:24

图 7-14　数据表 incremental_source

id	name	age	create_time
1	Isabella	18	2019-08-20 13:14:20
2	Jack	20	2019-08-20 13:14:21
3	Nicholas	22	2019-08-20 13:14:22
4	Jasmine	19	2019-08-20 13:14:23
5	Mia	20	2019-08-20 13:14:24

图 7-15　数据表 incremental_target

下面通过 Kettle 工具将数据表 incremental_source 中的数据增量加载到数据表 incremental_target 中，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 incremental_load，并添加“表输入”控件、“插入/更新”控件以及 Hop 跳连接线，具体效果如图 7-16 所示。

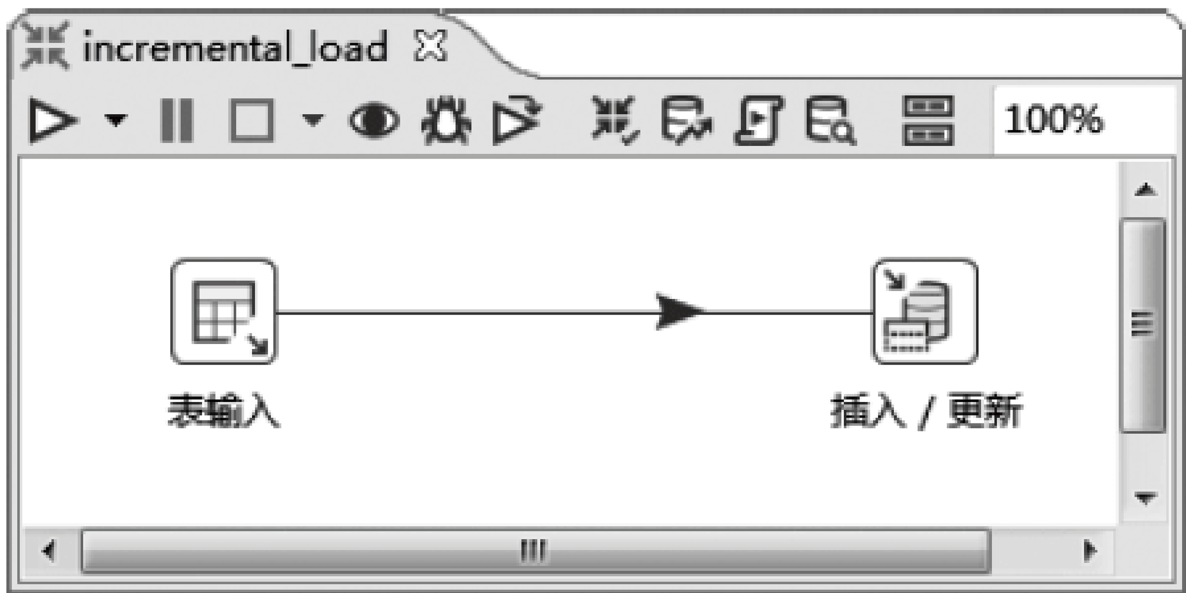


图 7-16　创建转换 incremental_load

2. 配置“表输入”控件

双击图 7-16 中的“表输入”控件，进入“表输入”界面，具体如图 7-17 所示。

在图 7-17 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 7-18 所示。

在图 7-17 的 SQL 框中编写查询数据表 incremental_source 数据的 SQL 语句，然后单击“预览”按钮，查看数据表 incremental_source 的数据是否成功从 MySQL 数据库中抽取到表输入流中，具体如图 7-19 和图 7-20 所示。

从图 7-20 中可以看出，数据表 incremental_source 的数据已经成功从 MySQL 数据库中抽取到表输入流中，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。需要注意的是，Kettle



图 7-17 “表输入”界面

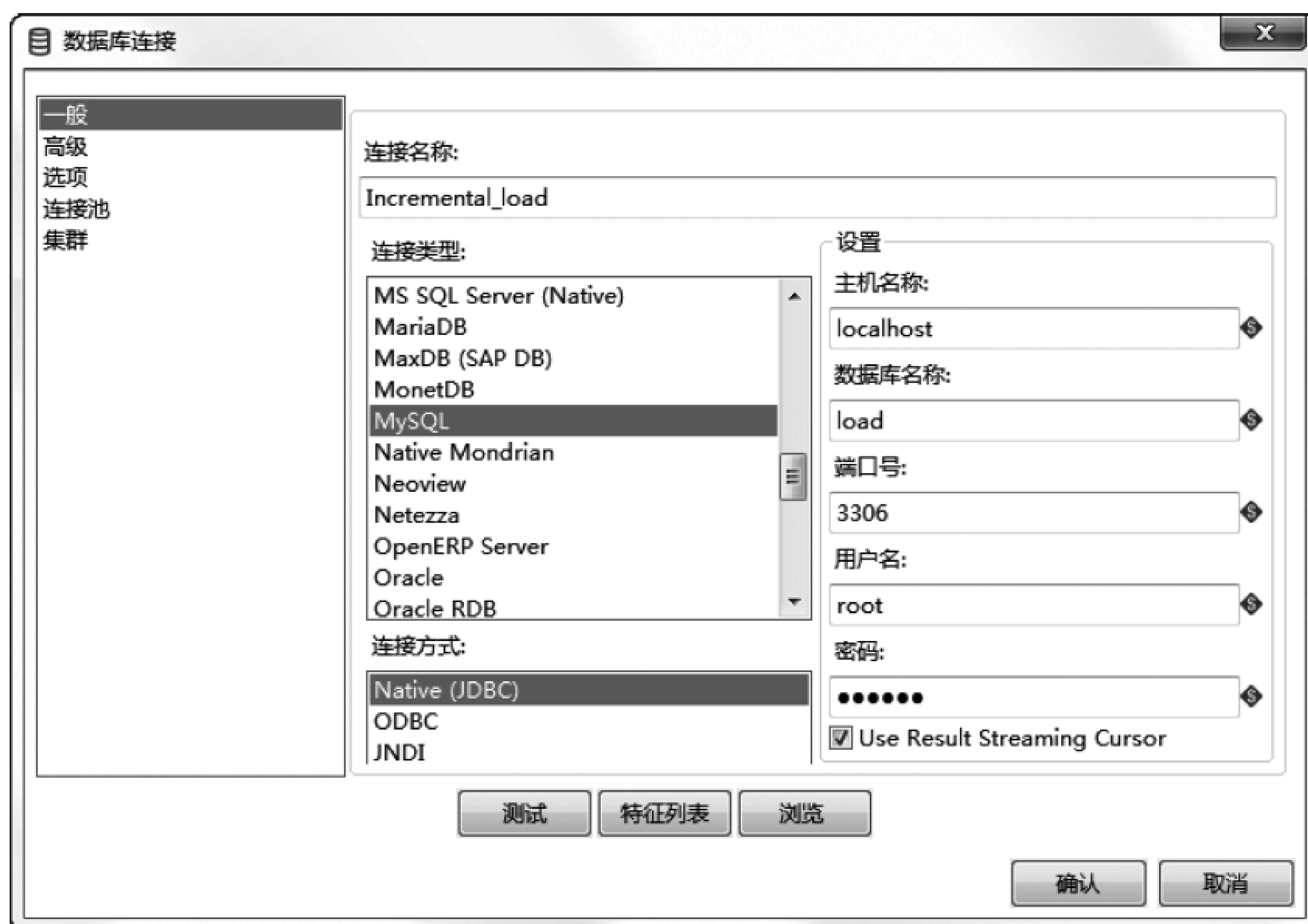


图 7-18 MySQL 数据库连接的配置

中读取的时间戳格式是精确度为毫秒级的，例如 yyyy/MM/dd hh：mm：ss. SSSSSSSSS 格式。

3. 配置“插入/更新”控件

双击图 7-16 中的“插入/更新”控件，进入“插入/更新”界面，具体如图 7-21 所示。

在图 7-21 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 7-22 所示。

单击图 7-21 中目标表处的“浏览”按钮，弹出“数据库浏览器”窗口，选择目标表 incremental_target，具体如图 7-23 所示。

在图 7-23 中单击“获取字段”按钮，用来指定查询数据需要的关键字，这里以比较数据



图 7-19 编写 SQL 语句

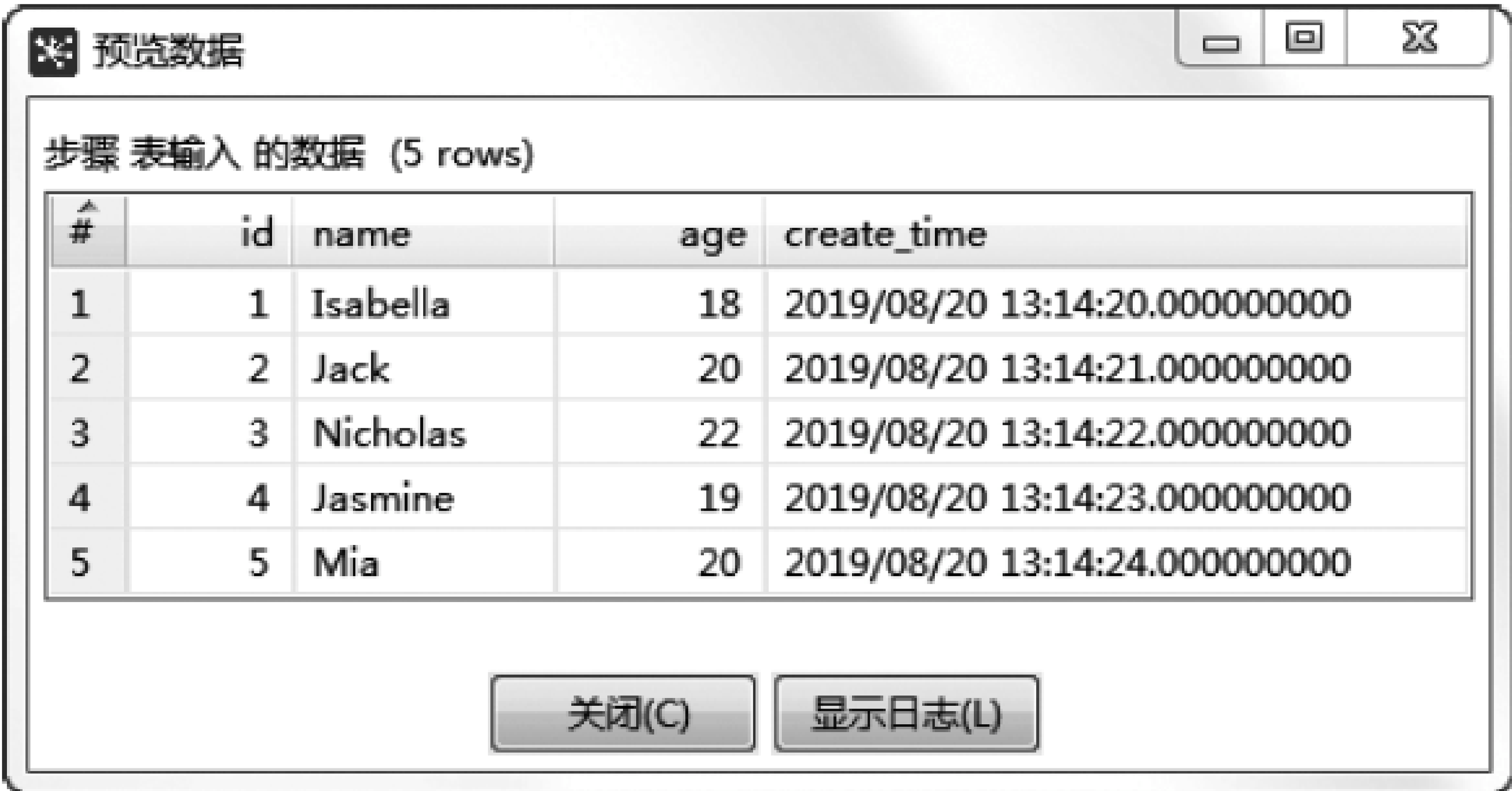


图 7-20 预览数据



图 7-21 “插入/更新”界面



图 7-22　MySQL 数据库连接的配置



图 7-23　选择目标表 incremental_target

表 incremental_target 的字段 id 与输入流里的字段 id 是否一致为关键条件，更新数据表中的其他字段数据；单击"获取和更新字段"按钮，用来指定需要更新的字段，具体如图 7-24 所示。

在图 7-24 中单击"确定"按钮，完成"插入/更新"控件的配置。

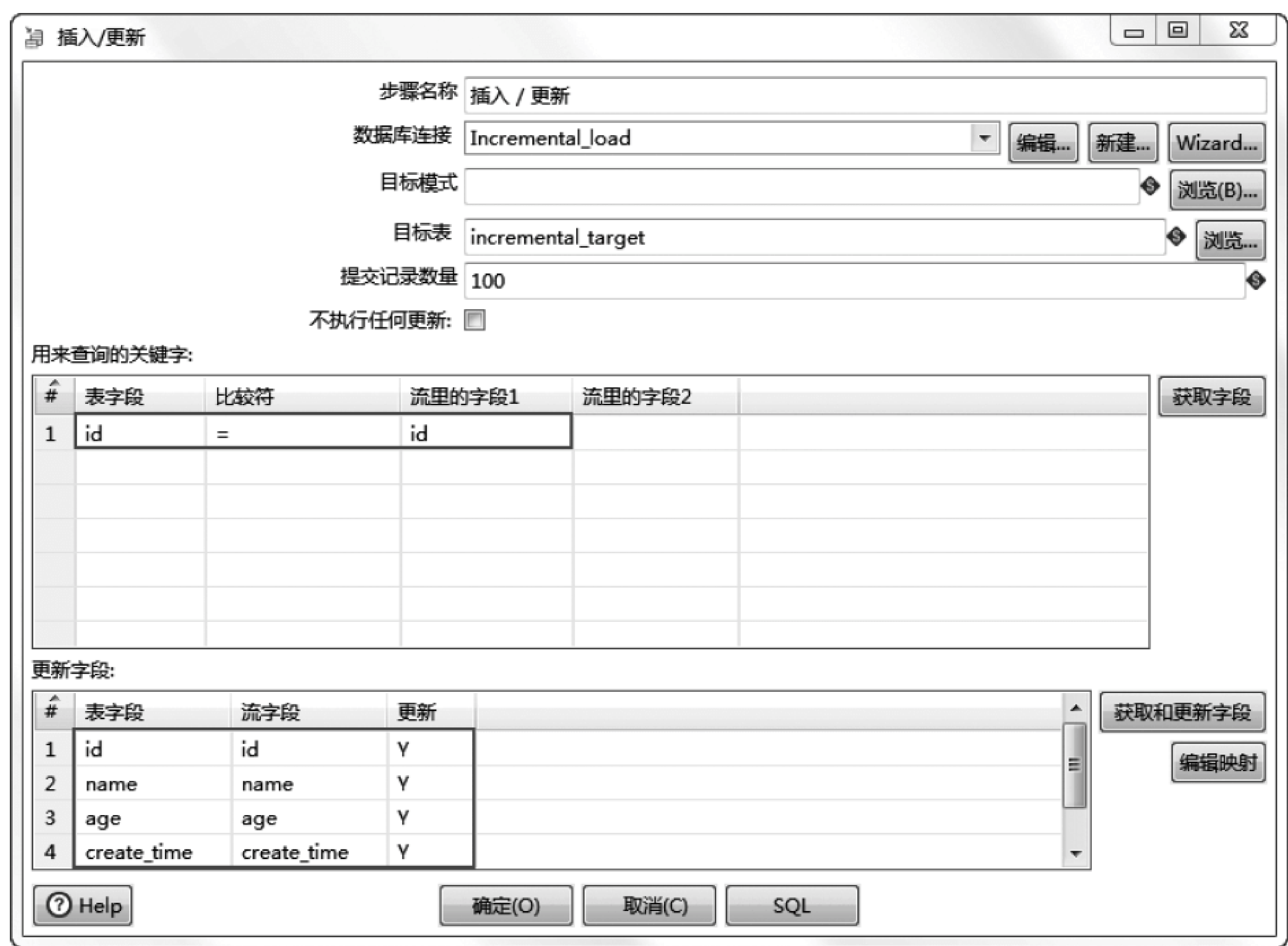


图 7-24 配置“插入/更新”控件

4．修改数据表 incremental_source 中的数据

在数据表 incremental_source 中新增一条 id 为 6、name 为 Mary、age 为 23 的数据；对数据表 incremental_source 中 id 为 2 的数据进行修改，即将这条数据的年龄(age)改为 25。修改后的数据表 incremental_source 如图 7-25 所示。

id	name	age	create_time
1	Isabella	18	2019-08-20 13:14:20
2	Jack	25	2019-08-21 14:04:21
3	Nicholas	22	2019-08-20 13:14:22
4	Jasmine	19	2019-08-20 13:14:23
5	Mia	20	2019-08-20 13:14:24
6	Mary	23	2019-08-20 13:14:25

图 7-25 修改后的数据表 incremental_source

5．运行转换 incremental_load

单击转换工作区顶部的 ▷ 按钮，运行创建的转换 incremental_load，实现将数据表 incremental_source 中的数据增量加载到数据表 incremental_target 中，具体如图 7-26 所示。

从图 7-26 中执行结果的“步骤度量”可以看出，“表输入”控件输入 6 条数据，并写入该控件；“插入/更新”控件从“表输入”控件中读取 6 条数据，并输入、写入到该控件，最终更新并输出至数据表 incremental_target 中。也就是说，数据表 incremental_source 中有 2 条数

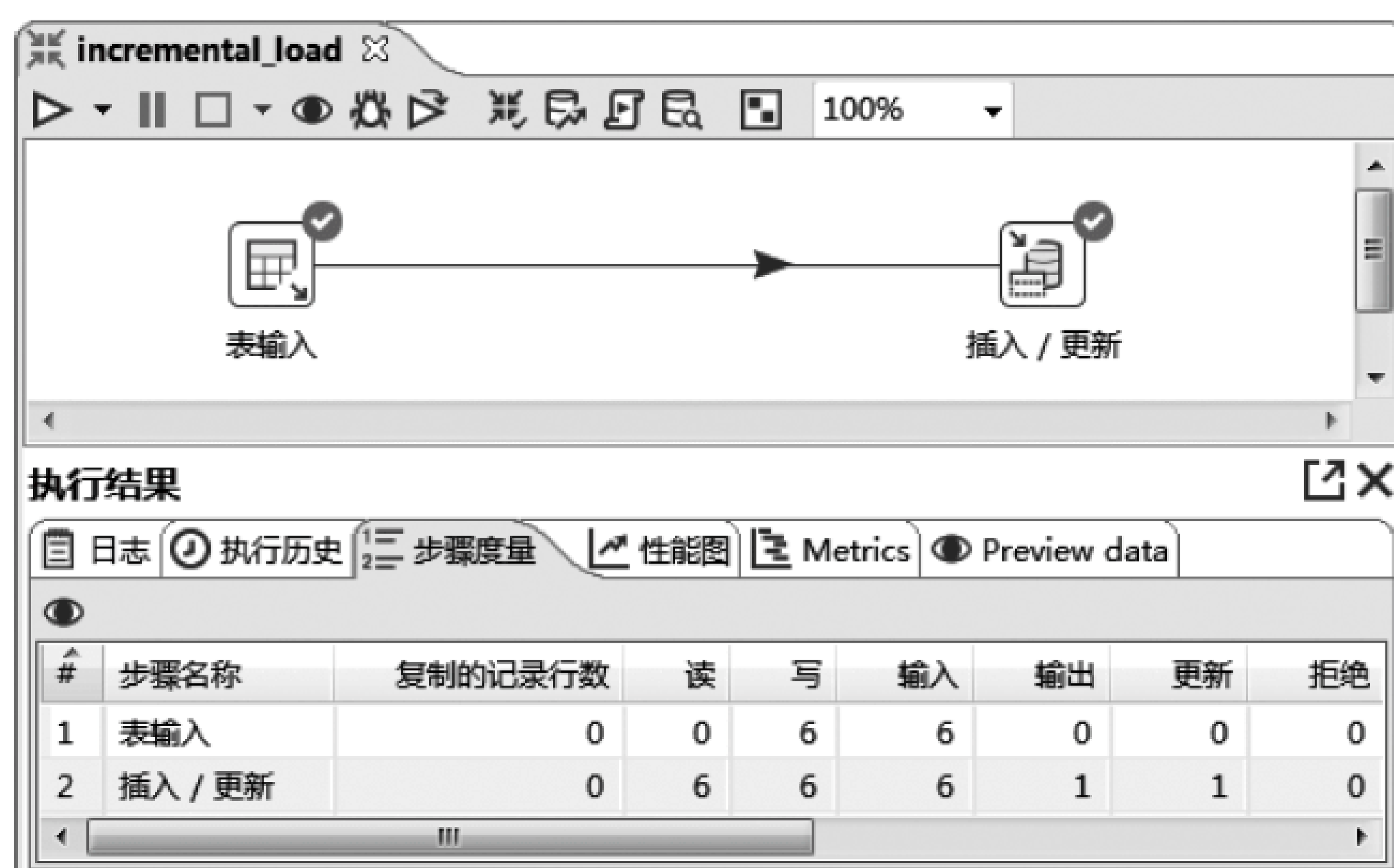


图 7-26 运行转换 incremental_load

据发生了变化，并将变化后的数据加载到了数据表 incremental_target 中。

6. 查看数据表 incremental_target 中的数据

通过 SQLyog 工具，查看数据表 incremental_target 中是否已成功加载到数据，查看结果如图 7-27 所示。

id	name	age	create_time
1	Isabella	18	2019-08-20 13:14:20
2	Jack	25	2019-08-21 14:04:21
3	Nicholas	22	2019-08-20 13:14:22
4	Jasmine	19	2019-08-20 13:14:23
5	Mia	20	2019-08-20 13:14:24
6	Mary	23	2019-08-20 13:14:25

图 7-27 数据表 incremental_target

从图 7-27 中可以看出，数据表 incremental_target 中已经加载到数据，说明我们通过 Kettle 工具成功实现了将数据表 incremental_source 中的数据增量加载到数据表 incremental_target 中。

7.2 数据的批量加载

通常，对于几千条甚至几十万条记录的数据迁移而言，采取 DML(数据操纵语言)的 INSERT 语句能够很好地将数据迁移到目标数据库中。然而，当数据迁移量过于庞大时，就不能使用 INSERT 语句了，因为执行 INSERT、UPDATE 以及 DELETE 语句的操作都会生成事物日志，事物日志的生成会减慢加载的速度，故需要针对数据采取批量加载操作。

假设有一个 CSV 格式的微博用户信息文件 weibo_user. csv，其中包含用户 id、用户名称、用户性别、用户简介等字段。文件 weibo_user. csv 的具体内容如图 7-28 所示(这里只截取了部分数据)。

下面通过 Kettle 工具将文件 weibo_user. csv 中的数据批量加载到数据表 weibo_user 中，具体实现步骤如下。

	A	B	C	D	E	F
1	user_id	user_name	gender	message	post_num	follower_num
2	1041514813	james1002	male	江宁波　简介：诚信为本，站的高才能看的远。快乐开心每一天。	1557	1421
3	1046222077	郑钧	male	悲歌欢唱　个性域名：zhengjun　博客地址：http://blog.s	1265	1280571
4	1087770692	陈坤	male	简介：莫失己道，莫扰他心。　个性域名：chenkun　标签	4461	74798328
5	1092538373	CCTV天下足球	male	下足球》，最纯粹的足球，最高级的享受！播出时间：每周一晚19:	1222	516668
6	1114281232	菩提2589	male	湖南长沙　简介：若你安好，便是晴天　标签　美女	333	199
7	1155631071	若水团	female	无来者的帮派:http://t.cn/8kI0tCF欢迎乱入!参团加旺旺18群：	10959	17399
8	1165712932	演员王澜	female	魁)(爱情二十年)(东方朔)(别拿自己不当干部)(四世同堂)(大时代)	5221	22572
9	1195210033	佟大为	male	@tongdaweistudio.com　个性域名：tongdawei　博客地址：	2668	16748591
10	1195264701	双双留头发	female	北京东城区　12月16日　简介：可爱温柔的幼儿园老师！	603	141
11	1197106530	baby俊良	male	四川宜宾　1987年2月3日	3890	305
12	1197161814	李开复	male	1961年12月3日　简介：创新工场CEO，媒体联系：press@chuangx	13970	50965446
13	1212812142	文章同學	male	ang626　博客地址：http://blog.sina.com.cn/wenzhang626	2213	55260225
14	1223178222	胡歌	male	完　简介：一呼吸一天地　个性域名：hu_ge　博客地址：h	2945	29063431
15	1223237202	华生2010	male	经济学家、东南大学教授　博客地址：http://blog.sina.com.c	858	451808
16	1223762662	林心如	female	9@gmail.com　个性域名：linxinru　博客地址：http://bl	6052	59348268
17	1226318347	育儿专家大百科	male	人，为父母们提供科学的育儿经验。Q:1278417449　标签　育儿	1354	1003824
18	1228486722	王珞丹	female	事請找@王珞丹工作室　个性域名：wangluodan　博客地址：h	1859	34538704
19	1230663070	唐嫣	female	个性域名：tangyan　博客地址：http://blog.sina.com.cn,	2470	29440315
20	1234552257	成龙	male	echan.com　个性域名：jackiechan　博客地址：http://bl	765	24246519

weibo_user

图 7-28　文件 weibo_user.csv 的具体内容

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 batch_load，并添加“CSV 文件输入”控件、“表输出”控件以及 Hop 跳连接线，具体效果如图 7-29 所示。

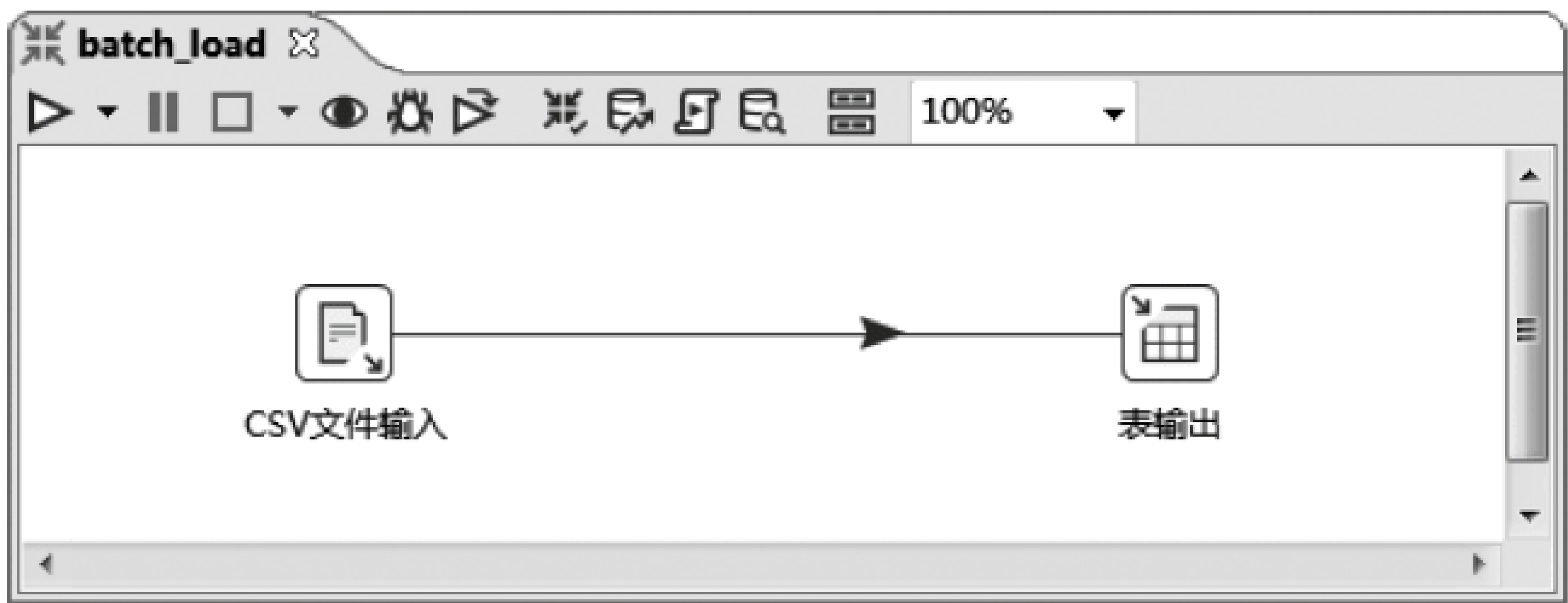


图 7-29　创建转换 batch_load

2. 配置“CSV 文件输入”控件

双击图 7-29 中的“CSV 文件输入”控件，进入“CSV 文件输入”界面，具体如图 7-30 所示。

在图 7-30 中的“文件名”处单击“浏览”按钮，选择要抽取的 CSV 文件 weibo_user.csv；单击“获取字段”按钮，让 Kettle 自动检索 CSV 文件，并对文件中字段的类型、格式、长度、精度等属性进行解析，具体效果如图 7-31 所示。

在图 7-31 中单击“预览”按钮，查看 CSV 文件 weibo_user.csv 的数据是否抽取到 CSV 文件输入流中，具体效果如图 7-32 所示。

从图 7-32 中可以看出，文件 weibo_user.csv 的数据已经成功抽取到 CSV 文件输入流中，单击“关闭”→“确定”按钮，完成“CSV 文件输入”控件的配置。

3. 配置“表输出”控件

双击图 7-29 中的“表输出”控件，进入“表输出”界面，具体如图 7-33 所示。



图 7-30 “CSV 文件输入”界面



图 7-31 配置“CSV 文件输入”控件

在图 7-33 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 7-34 所示。

单击图 7-33 中目标表右侧的“浏览”按钮，选择输出的目标表，即数据表 weibo_user（该表须提前创建，且表结构须根据文件 weibo_user.csv 中数据的字段和数据类型创建，这里不演示）；勾选“指定数据库字段”复选框，用于将数据表 weibo_user 的字段与文件 weibo_

预览数据

步骤 CSV文件输入 的数据 (99 rows)

#	user_id	user_name	gender	message	post_num	follower_num
1	1041514813	james1002	male	浙江宁波　简介：诚信为本，站的高才能看的远。快...	1557	1421
2	1046222077	郑钧	male	北京东城区　简介：且把悲歌欢唱　个性域名：zhen...	1265	1280571
3	1087770692	陈坤	male	重庆　简介：莫失己道，莫扰他心。　个性域名：ch...	4461	74798328
4	1092538373	CCTV天下足球	male	简介：《天下足球》，最纯粹的足球，最高级的享受...	1222	516668
5	1114281232	菩提2589	male	湖南长沙　简介：若你安好，便是晴天　标签　美女	333	199
6	1155631071	若水团	female	海外其他　恋爱中　狮子座　简介：方便得前无古人...	10959	17399
7	1165712932	演员王澜	female	北京朝阳区　毕业于　中央戏剧学院　天秤座　简介...	5221	22572
8	1195210033	佟大为	male	北京东城区　简介：任何业务请与我的经纪人联系，...	2668	16748591
9	1195264701	双双留头发	female	北京东城区　12月16日　简介：可爱温柔的幼儿园老...	603	141
10	1197106530	baby俊良	male	四川宜宾　1987年2月3日	3890	305
11	1197161814	李开复	male	北京东城区　公司　创新工场　1961年12月3日　简...	13970	50965446
12	1212812142	文章同學	male	北京朝阳区　个性域名：wenzhang626　博客地址...	2213	55260225
13	1223178222	胡歌	male	上海徐汇区　毕业于　上海戏剧学院　简介：一呼吸...	2945	29063431
14	1223237202	华生2010	male	其他　简介：经济学家、东南大学教授　博客地址：...	858	451808
15	1223762662	林心如	female	台湾　简介：工作聯繫~lisa谭小姐tanlisa0829@gm...	6052	59348268
16	1226318347	育儿专家大...	male	天津和平区　5月1日　简介：专业的育儿智囊团队，...	1354	1003824
17	1228486722	王珞丹	female	北京　简介：博主衹在這裡瞎胡鬧，正經事請找@王...	1859	34538704
18	1230663070	唐嫣	female	上海　简介：lovetiffanytang@foxmail.com糖糖.TIF...	2470	29440315
19	1234552257	成龙	male	北京　简介：所有业務及邀请，請联系：jcgroup@ja...	765	24246519
20	1235527153	翱翔6688	male	其他　1980年6月8日　简介：陌路相逢，以文会友；...	1078	31396
21	1249193625	乐嘉	male	上海黄浦区　1975年5月16日　简介：官网www.fpa...	3133	41915592

关闭(C)　显示日志(L)

图 7-32　预览数据

表输出

步骤名称　表输出

数据库连接　编辑...　新建...　Wizard...

目标模式　浏览(B)...

目标表　浏览(B)...

提交记录数量　1000

裁剪表

忽略插入错误

指定数据库字段

主选项　数据库字段

表分区数据

Help　确定(O)　取消(C)　SQL

图 7-33　"表输出"界面

user.csv 中的字段进行匹配；勾选"使用批量插入"复选框，用于批量加载数据至目标表中，具体如图 7-35 所示。

在图 7-35 中选择"数据库字段"选项卡，具体如图 7-36 所示。

在图 7-36 中单击"输入字段映射"按钮，弹出"映射匹配"对话框，具体如图 7-37 所示。

在图 3-37 中依次选中"源字段"中的字段和"目标字段"中的字段，再单击 Add 按钮，将



图 7-34 MySQL 数据库连接的配置



图 7-35 指定输出的目标表

一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 7-38 所示。

在图 7-38 中单击“映射匹配”对话框中的“确定”按钮，“表输出”控件的配置效果如图 7-39 所示。

在图 7-39 中单击“确定”按钮，完成“表输出”控件的配置。



图 7-36　“数据库字段”选项卡



图 7-37　“映射匹配”对话框



图 7-38　设置映射匹配

4. 运行转换 batch_load

单击转换工作区顶部的▷按钮，运行创建的转换 batch_load，实现将 CSV 文件 weibo_user.csv 中的数据批量加载到数据表 weibo_user 中，具体如图 7-40 所示。

从图 7-40 中执行结果的“步骤度量”可以看出，“CSV 文件输入”控件输入 100 条数据，



图 7-39 “表输出”控件的配置效果

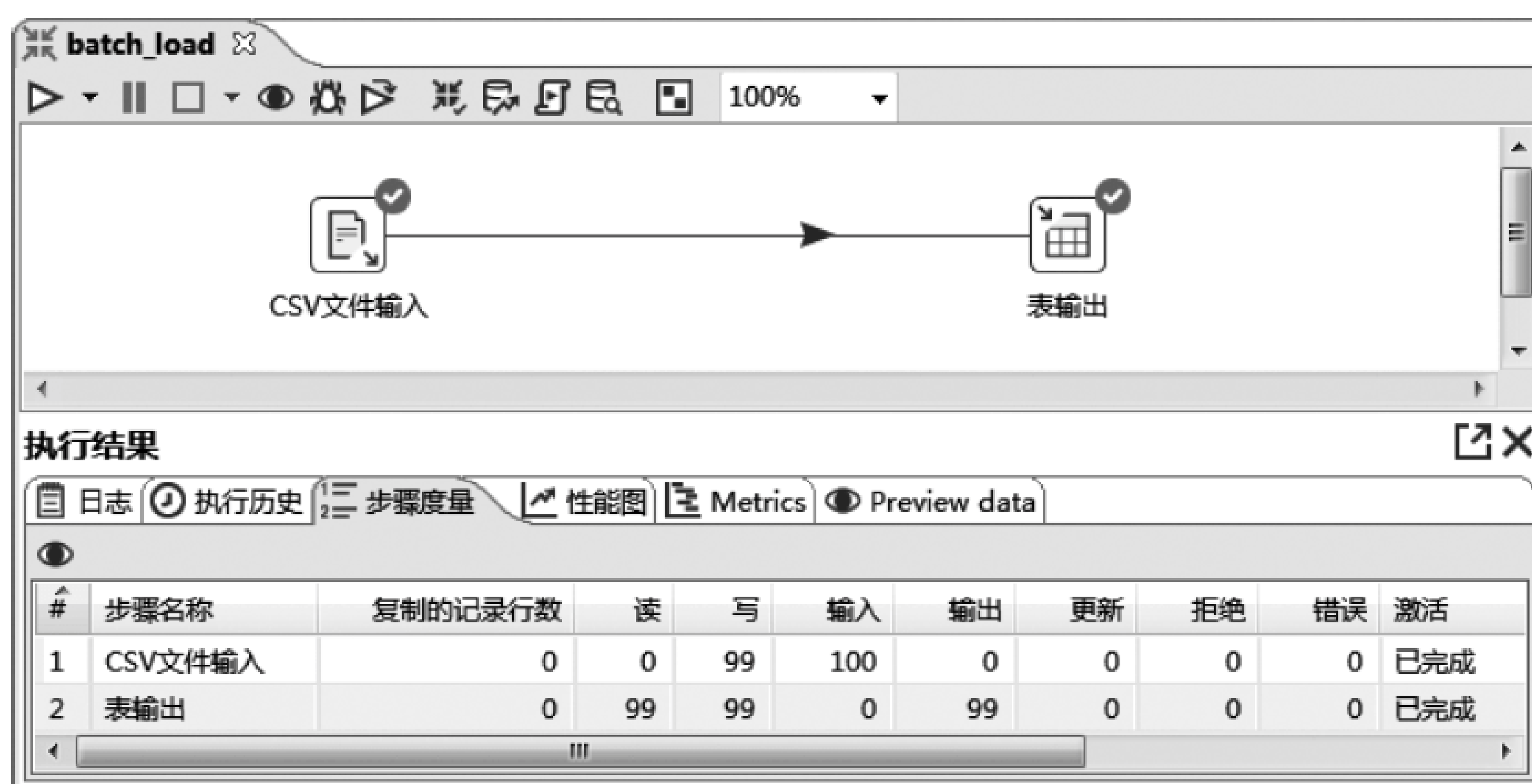


图 7-40 运行转换 batch_load

写入该控件 99 条数据(1 条表头数据未写入);“表输出”控件从“CSV 文件输入”控件中读取 99 条数据,并写入该控件,最终进行输出。

5. 查看数据表 weibo_user 中的数据

通过 SQLyog 工具,查看数据表 weibo_user 是否已成功加载到数据,查看结果如图 7-41 所示。

从图 7-41 中可以看出,数据表 weibo_user 中已经加载到数据,说明我们通过 Kettle 工具成功实现了将 CSV 文件 weibo_user.csv 中的数据批量加载到数据表 weibo_user 中。

user_id	user_name	gender	message	post_num	follower_num
1041514813	james1002	male	浙江宁波　简介：诚信为本，站的高才能看的远。	1557	1421
1046222077	郑钧	male	北京东城区　简介：且把悲歌欢唱　个性域名：	1265	1280571
1087770692	陈坤	male	重庆　简介：莫失己道，莫扰他心。　个性域名	4461	74798328
1092538373	CCTV天下足球	male	简介：《天下足球》，最纯粹的足球，最高级的享	1222	516668
1114281232	菩提2589	male	湖南长沙　简介：若你安好，便是晴天　标签	333	199
1155631071	若水团	female	海外其他　恋爱中　狮子座　简介：方便得前	10959	17399
1165712932	演员王澜	female	北京朝阳区　毕业于　中央戏剧学院　天秤座	5221	22572
1195210033	佟大为	male	北京东城区　简介：任何业务请与我的经纪人联	2668	16748591
1195264701	双双留头发	female	北京东城区　12月16日　简介：可爱温柔的幼儿	603	141
1197106530	baby俊良	male	四川宜宾　1987年2月3日	3890	305
1197161814	李开复	male	北京东城区　公司　创新工场　1961年12月3日	13970	50965446
1212812142	文章同學	male	北京朝阳区　个性域名：　wenzhang626　博客	2213	55260225
1223178222	胡歌	male	上海徐汇区　毕业于　上海戏剧学院　简介：	2945	29063431
1223237202	华生2010	male	其他　简介：经济学家、东南大学教授　博客地	858	451808
1223762662	林心如	female	台湾　简介：工作聯繫~lisa谭小姐tanlisa0829	6052	59348268
1226318347	育儿专家大百	male	天津和平区　5月1日　简介：专业的育儿智囊团	1354	1003824
1228486722	王珞丹	female	北京　简介：博主祇在這裡瞎胡鬧，正經事請找	1859	34538704
1230663070	唐嫣	female	上海　简介：lovetiffanytang@foxmail.com糖糖	2470	29440315
1234552257	成龙	male	北京　简介：所有业務及邀请，請联系：jcgrour	765	24246519

图 7-41　数据表 weibo_user

7.3　本章小结

本章主要讲解数据加载的相关知识，包括数据的全量加载、增量加载以及批量加载。希望读者通过本章的学习，可以掌握数据的全量加载、增量加载以及批量加载的操作，实现将企业中清洗检验、转换后的高质量数据加载到目标数据库中，便于后续进行数据分析和数据挖掘。

7.4　本章习题

一、填空题

1. 数据的加载机制可以分为________和增量加载。
2. 增量加载是指目标表仅加载源数据表中________的数据。
3. 当数据迁移量过于庞大时，需要针对数据采取________操作。

二、操作题

1. 现有两张数据表，分别为 full_source 和 full_target，其中 full_source 为源数据表，full_target 为目标数据表。数据表 full_source 和 full_target 分别如图 7-42 和图 7-43 所示。

no	book_name	author	tag
1	小王子	圣艾克苏佩里	外国文学
2	解忧杂货店	东野圭吾	小说
3	白夜行	东野圭吾	小说
4	梦里花落知多少	郭敬明	小说
5	看见	柴静	随笔
6	围城	钱钟书	小说

图 7-42　数据表 full_source

no	book_name	author	tag
1	小王子	圣艾克苏佩里	外国文学
2	解忧杂货店	东野圭吾	小说
3	看见	柴静	随笔

图 7-43 数据表 full_target

通过使用 Kettle 工具,实现将数据表 full_source 中的数据全量加载到数据表 full_target 中。

2. 现有两张数据表,分别为 incremental_source 和 incremental_target,其中 incremental_source 为源数据表,incremental_target 为目标数据表。数据表 incremental_source 和 incremental_target 的表结构、数据都是相同的,具体如图 7-44 和图 7-45 所示。

id	name	age	create_time
1	Isabella	18	2019-08-20 13:14:20
2	Jack	20	2019-08-20 13:14:21
3	Nicholas	22	2019-08-20 13:14:22
4	Jasmine	19	2019-08-20 13:14:23
5	Mia	20	2019-08-20 13:14:24

图 7-44 数据表 incremental_source

id	name	age	create_time
1	Isabella	18	2019-08-20 13:14:20
2	Jack	20	2019-08-20 13:14:21
3	Nicholas	22	2019-08-20 13:14:22
4	Jasmine	19	2019-08-20 13:14:23
5	Mia	20	2019-08-20 13:14:24

图 7-45 数据表 incremental_target

通过使用 Kettle 工具,实现将数据表 incremental_source 中的数据增量加载到数据表 incremental_target 中,即将数据表 incremental_source 中新增、变化的数据加载到数据表 incremental_target 中。

(注:新增数据,在数据表 incremental_source 中新增一条 id 为 6、name 为 Mary、age 为 23 的数据;变化数据,修改数据表 incremental_source 中 id 为 2 的数据,将该条数据的 age 值改为 25)。

第 8 章
综合案例——构建DVD租赁商店数据仓库

学习目标

（1）了解数据库 sakila 中的数据表

（2）理解数据仓库 sakila_dw 的架构设计

（3）熟悉 DVD 租赁商店的业务流程

（4）掌握构建 DVD 租赁商店数据仓库的具体实现

sakila 样本数据库是 MySQL 官方提供的一个模拟 DVD 租赁商店管理的数据库。本章将综合运用前面几章的知识，对数据库 sakila 中的数据进行清洗操作，从而构建一个 DVD 租赁商店数据仓库系统，即实现定期从源数据库 sakila 中抽取增量数据，转换成符合 DVD 租赁业务的数据，最后加载到目标数据仓库中。

8.1 案例概述

8.1.1 案例背景介绍

在日益激烈的商业竞争中，在线 DVD 租赁商店的决策者都迫切需要更加准确的经营决策信息。每个在线 DVD 租赁商店的数据都存储在数据库中，因此该数据库中拥有海量数据，不缺乏足够的信息，但是这些数据并不是经营决策需要的信息。虽然这些海量数据对于在线 DVD 租赁商店的运作非常有用，但是对于商业的战略决策和目标制定的作用是微乎其微的。

对于在线 DVD 租赁商店的决策者来说，他们需要从多个不同的商业角度观察数据，如从时间、电影、演员、用户等角度观察数据，并进行相关的分析得出决策，但是数据库中的数据不适合从多个角度进行分析，无法得出战略决策。然而，数据仓库支持复杂的分析操作，侧重于决策支持，并且还提供直观易懂的查询结果，因此我们需要基于数据库 sakila 创建一个 DVD 租赁商店数据仓库，并将 sakila 数据库中的数据加载到数据仓库中，以便于在线 DVD 租赁商店的决策者对数据进行分析得出商业决策。

8.1.2 数据仓库的架构模型

下面通过一张图描述 DVD 租赁商店数据仓库 sakila_dw 的架构模型，具体如图 8-1 所示。

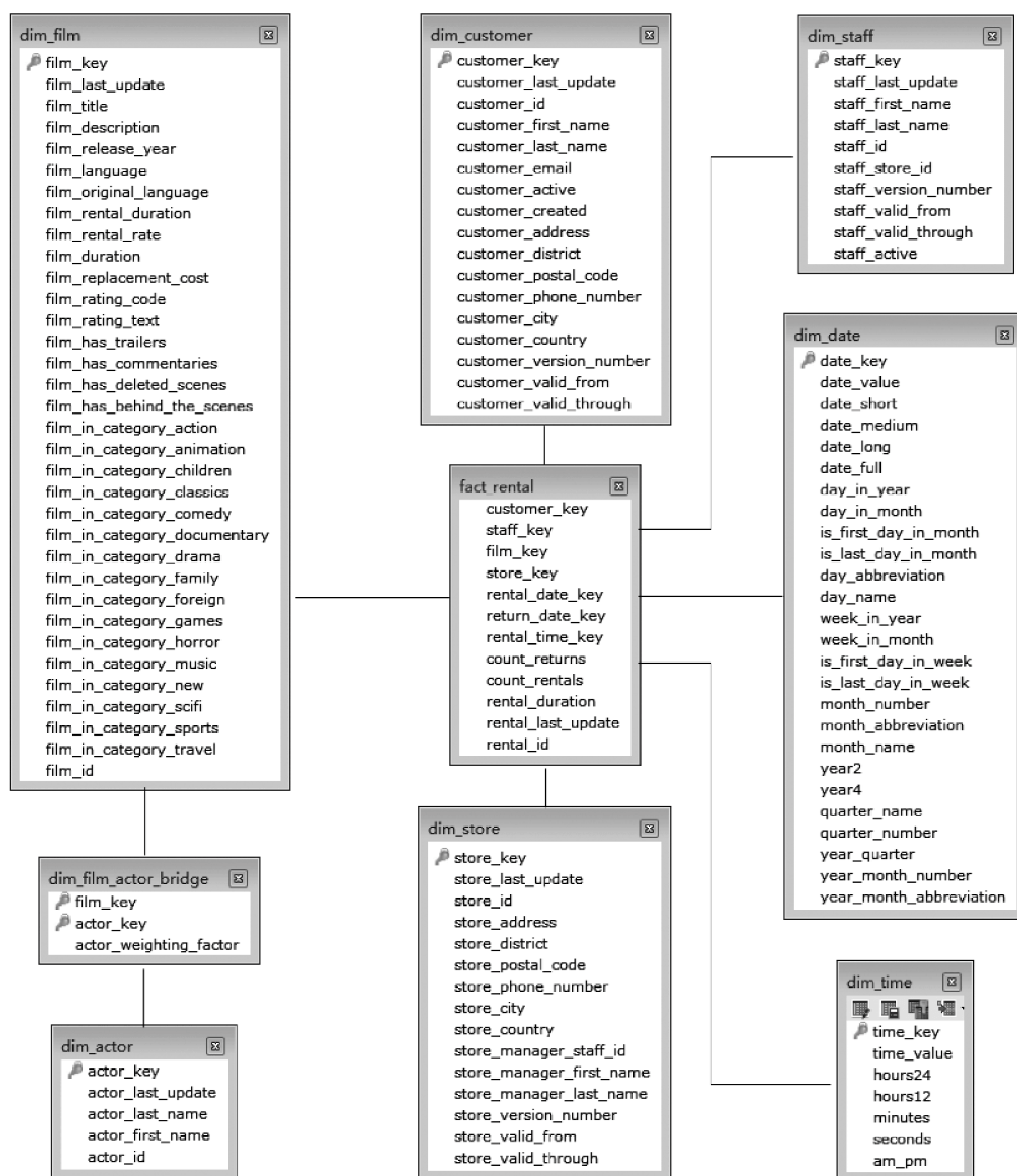


图 8-1 数据仓库 sakila_dw 的架构模型

从图 8-1 中可以看出,数据仓库 sakila_dw 的架构模型是一个星形模型,其中 dim_film 表、dim_customer 表、dim_actor 表、dim_store 表、dim_staff 表、dim_date 表、dim_film_actor_bridge 表以及 dim_time 表均为维度表;fact_rental 表为事实表。

8.1.3 数据仓库效果预览

通过 Kettle 工具,将 sakila 数据库中的数据加载至 sakila_dw 数据仓库中。数据仓库 sakila_dw 中的维度表和事实表中的部分数据,具体如图 8-2～图 8-9 所示。

actor_key	actor_last_update	actor_last_name	actor_first_name	actor_id
401	2006-02-15 04:34:33	GUINESS	PENELOPE	1
402	2006-02-15 04:34:33	WAHLBERG	NICK	2
403	2006-02-15 04:34:33	CHASE	ED	3
404	2006-02-15 04:34:33	DAVIS	JENNIFER	4
405	2006-02-15 04:34:33	LOLLOBRIGIDA	JOHNNY	5
406	2006-02-15 04:34:33	NICHOLSON	BETTE	6
407	2006-02-15 04:34:33	MOSTEL	GRACE	7
408	2006-02-15 04:34:33	JOHANSSON	MATTHEW	8
409	2006-02-15 04:34:33	SWANK	JOE	9
410	2006-02-15 04:34:33	GABLE	CHRISTIAN	10

图 8-2　维度表 dim_actor 中的数据

cust...	customer_last_up...	custom...	custo...	customer_la...	customer...	customer
4791	2006-02-15 04:57:20	597	FREDDIE	DUGGAN	FREDDIE.DUGG	Yes
4792	2006-02-15 04:57:20	598	WADE	DELVALLE	WADE.DELVALL	Yes
4793	2006-02-15 04:57:20	599	AUSTIN	CINTRON	AUSTIN.CINTR	Yes
4790	2006-02-15 04:57:20	596	ENRIQUE	FORSYTHE	ENRIQUE.FORS	Yes
4789	2006-02-15 04:57:20	595	TERRENCE	GUNDERSON	TERRENCE.GUN	Yes
4786	2006-02-15 04:57:20	592	TERRANCE	ROUSH	TERRANCE.ROU	No
4787	2006-02-15 04:57:20	593	RENE	MCALISTER	RENE.MCALIST	Yes
4788	2006-02-15 04:57:20	594	EDUARDO	HIATT	EDUARDO.HIAT	Yes
4785	2006-02-15 04:57:20	591	KENT	ARSENAULT	KENT.ARSENAU	Yes

图 8-3　维度表 dim_customer 中的数据

date_key	date_v...	date_short	date_medium	date_long	date_full
20091228	2009-12-28	12/28/09	Dec 28, 2009	December 28, 2009	Monday, December 28, 20(
20091227	2009-12-27	12/27/09	Dec 27, 2009	December 27, 2009	Sunday, December 27, 20(
20091226	2009-12-26	12/26/09	Dec 26, 2009	December 26, 2009	Saturday, December 26, :
20091225	2009-12-25	12/25/09	Dec 25, 2009	December 25, 2009	Friday, December 25, 20(
20091224	2009-12-24	12/24/09	Dec 24, 2009	December 24, 2009	Thursday, December 24, :
20091223	2009-12-23	12/23/09	Dec 23, 2009	December 23, 2009	Wednesday, December 23,
20091222	2009-12-22	12/22/09	Dec 22, 2009	December 22, 2009	Tuesday, December 22, 2(
20091221	2009-12-21	12/21/09	Dec 21, 2009	December 21, 2009	Monday, December 21, 20(
20091220	2009-12-20	12/20/09	Dec 20, 2009	December 20, 2009	Sunday, December 20, 20(

图 8-4　维度表 dim_date 中的数据

film_key	film_last_update	film_title	film_description		fil...	film...	film
2907	2006-02-15 05:03:42	TRANSLATION SUMMI	A Touch...	93B	2006	English	Not A
2906	2006-02-15 05:03:42	TRAMP OTHERS	A Brill...	87B	2006	English	Not A
2905	2006-02-15 05:03:42	TRAINSPOTTING STI	A Fast-...	90B	2006	English	Not A
2904	2006-02-15 05:03:42	TRAIN BUNCH	A Thril...	90B	2006	English	Not A
2903	2006-02-15 05:03:42	TRAFFIC HOBBIT	A Amazi...	102B	2006	English	Not A
2902	2006-02-15 05:03:42	TRADING PINOCCHI(	A Emoti...	117B	2006	English	Not A
2901	2006-02-15 05:03:42	TRACY CIDER	A Touch...	101B	2006	English	Not A
2900	2006-02-15 05:03:42	TOWN ARK	A Awe-I...	98B	2006	English	Not A
2899	2006-02-15 05:03:42	TOWERS HURRICANE	A Fatef...	85B	2006	English	Not A

图 8-5　维度表 dim_film 中的数据

film_key	actor_key	actor_weighting_factor
1001	210	0.20
1001	230	0.20
1001	253	0.20
1001	362	0.20
1001	398	0.20
1002	285	0.50
1002	360	0.50
1003	219	0.33
1003	264	0.33
1004	241	0.50

图 8-6　维度表 dim_film_actor_bridge 中的数据

staff_key	staff_last_u...	staff_f...	staff_...	sta...	staff_...	staff_v...	staff_..
15	1970-01-01 00:00	Jon	Stephens	2	2	1	1900-01-
16	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	1	(NULL)
14	1970-01-01 00:00	Mike	Hillyer	1	1	1	1900-01-
13	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	1	(NULL)
(Auto)	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

图 8-7 维度表 dim_staff 中的数据

time_key	time_value	hours24	hours12	minutes	seconds	am_pm
0	00:00:00	0	0	0	0	AM
1	00:00:01	0	0	0	1	AM
2	00:00:02	0	0	0	2	AM
3	00:00:03	0	0	0	3	AM

图 8-8 维度表 dim_time 中的数据

cust...	s..	film_key	s...	renta...	return_d...	rent...	co...	c..	ren...	rental_las
4748	15	2001	7	20050821	0	3032	0	1	(NULL)	2006-02-23
2908	10	1135	5	20050529	20050602	210032	1	1	324900	2006-02-15
2419	10	1567	4	20050529	20050607	210722	1	1	702900	2006-02-15
2745	9	1741	5	20050529	20050604	213112	1	1	523080	2006-02-15

图 8-9 事实表 fact_rental 中的数据

8.2 数据准备

8.2.1 数据库 sakila 的下载和安装

可以从 MySQL 的官网下载数据库 sakila 的建库脚本，若是在 Windows 环境下安装数据库 sakila，则下载名称为 sakila-db. zip 的压缩包文件；若是在 Linux 环境下安装数据库 sakila，则需要下载名称为 sakila-db. tar. gz 的压缩包文件。

本书下载的是名称为 sakila-db. zip 的压缩包文件，该压缩包文件中包含 3 个文件，分别是文件 sakila. mwb、文件 sakila-data. sql 和文件 sakila-schema. sql。其中，文件 sakila. mwb 是一个 MySQL Workbench 数据模型，用于查看数据库结构；文件 sakila-data. sql 是用于创建数据库 sakila 的数据；文件 sakila-schema. sql 是用于创建数据库 sakila 的数据结构。

数据库 sakila 下载完成后，直接解压压缩包，然后使用 MySQL 图形化管理软件 SQLyog 先运行脚本文件 sakila-schema. sql 创建数据库 sakila 和数据表，再运行脚本文件 sakila-data. sql 向数据库 sakila 中的数据表加载数据，最后刷新并查看数据库 sakila 中的数据表及数据表中的数据，若数据表中均含有数据，则说明安装数据库 sakila 成功，否则说明安装不成功，需要重新解压安装。

需要注意的是，安装数据库 sakila 之前需要下载并安装 MySQL 关系型数据库，并且版本不可低于 5.0，本书使用的是 MySQL 8.0.16 版本。关于 MySQL 数据库的下载安装，这里不作详细介绍，读者可自行下载安装。

8.2.2 数据库 sakila 简介

数据库 sakila 中一共含有 16 张数据表，分别是 actor（演员）表、address（地址）表、

category(类别)表、city(城市)表、country(国家)表、customer(顾客)表、film(电影)表、film_actor(演员所属电影)表、film_category(电影所属的类别)表、film_text(电影描述)表、inventory(库存)表、language(语言)表、payment(付款)表、rental(租赁)表、staff(工作人员)表以及 store(商店)表,这 16 张表在设计上有一些相同之处,具体如下。

- 每张数据表都有自增主键列,列名采用“表名_id”的格式命名。
- 每张数据表的外键约束都引用主键,外键的名称与主键的名称相同。
- 每张数据表中均有一个名称为 last_update 的列,数据类型为 TIMESTAMP(时间戳),主要用来记录增加或更新数据时的时间。

下面通过一张图描述数据库 sakila 中数据表之间的关系,如图 8-10 所示。

图 8-10　数据库 sakila 中数据表之间的关系

在图 8-10 中,为了便于了解数据库 sakila 中 16 张数据表的主从关系,可以将数据表进行分类,分为电影类、商店类、客户类以及区域类,其中电影类数据表包含 film 表、file_category 表、category 表、file_actor 表、actor 表、language 表、film_text 表;商店类数据表包含 store 表、staff 表、inventory 表;客户类数据表包含 customer 表、rental 表及 payment 表;区域类数据表包含 country 表、city 表以及 address 表。

8.2.3 数据表简介

通过 8.2.2 节我们了解到数据库 sakila 中 16 张数据表的基本信息，下面基于这些数据表进行详细介绍。

1. 数据表 film

数据表 film 用于存储电影的基本信息及相关介绍的数据，该数据表各个字段的含义见表 8-1。

表 8-1 数据表 film

字段名称	数据类型	相关说明
film_id	smallint	主键(电影 id)
title	varchar	电影名称
description	text	电影描述
release_year	year	上映年份
language_id	tinyint	语言 id
original_language_id	tinyint	原版语言 id
rental_duration	tinyint	租赁时长
rental_rate	decimal	电影租赁费
length	smallint	电影时长
replacement_cost	decimal	替换成本
rating	enum	评分
special_features	set	特色
last_update	timestamp	最后更新时间

2. 数据表 file_category

数据表 file_category 用于存储定义电影 id 和所属电影类别 id 的数据，该数据表各个字段的含义见表 8-2。

表 8-2 数据表 file_category

字段名称	数据类型	相关说明
film_id	smallint	主键(电影 id)
category_id	tinyint	外键(电影类别 id)
last_update	timestamp	最后更新时间

3. 数据表 category

数据表 category 用于存储电影类别名称和所属类别 id 的数据，该数据表各个字段的含义见表 8-3。

表 8-3　数据表 category

字段名称	数据类型	相关说明
category_id	tinyint	主键(电影类别 id)
name	varchar	电影类别名称
last_update	timestamp	最后更新时间

4. 数据表 film_actor

数据表 film_actor 用于存储定义演员 id 和所属电影 id 的数据，该数据表各个字段的含义见表 8-4。

表 8-4　数据表 film_actor

字段名称	数据类型	相关说明
actor_id	smallint	主键(演员 id)
film_id	smallint	外键(电影 id)
last_update	timestamp	最后更新时间

5. 数据表 actor

数据表 actor 用于存储演员 id 对应的姓氏和名字数据，该数据表各个字段的含义见表 8-5。

表 8-5　数据表 actor

字段名称	数据类型	相关说明
actor_id	smallint	主键(演员 id)
first_name	varchar	演员名字
last_name	varchar	演员姓氏
last_update	timestamp	最后更新时间

6. 数据表 language

数据表 language 用于存储电影语言 id 和对应的语言名称数据，该数据表各个字段的含义见表 8-6。

表 8-6　数据表 language

字段名称	数据类型	相关说明
language_id	tinyint	主键(电影语言 id)
name	char	电影语言名称
last_update	timestamp	最后更新时间

7. 数据表 film_text

数据表 film_text 用于存储电影 id 和对应的电影名称及简述的数据，该数据表各个字段的含义见表 8-7。

表 8-7 数据表 film_text

字段名称	数据类型	相关说明
film_id	smallint	主键(电影 id)
title	varchar	电影名称
description	text	电影简述

8. 数据表 store

数据表 store 用于存储商店 id 和对应管理人员 id 以及商店地址 id 的数据，该数据表各个字段的含义见表 8-8。

表 8-8 数据表 store

字段名称	数据类型	相关说明
store_id	tinyint	主键(商店 id)
manager_staff_id	tinyint	管理人员 id
address_id	smallint	商店地址 id
last_update	timestamp	最后更新时间

9. 数据表 staff

数据表 staff 用于存储员工的基本信息及员工所属商店的数据，该数据表各个字段的含义见表 8-9。

表 8-9 数据表 staff

字段名称	数据类型	相关说明
staff_id	tinyint	主键(员工 id)
first_name	varchar	员工名字
last_name	varchar	员工姓氏
address_id	smallint	地址 id
picture	blob	照片
email	varchar	邮箱
store_id	tinyint	商店 id
active	tinyint	在职

续表

字段名称	数据类型	相关说明
username	varchar	用户名
password	varchar	密码
last_update	timestamp	最后更新时间

10. 数据表 inventory

数据表 inventory 用于存储库存编号对应的电影 id 和商店 id 数据，该数据表各个字段的含义见表 8-10。

表 8-10　数据表 inventory

字段名称	数据类型	相关说明
inventory_id	mediumint	主键(库存编号)
film_id	smallint	电影 id
store_id	tinyint	商店 id
last_update	timestamp	最后更新时间

11. 数据表 customer

数据表 customer 用于存储顾客的基本信息数据，该数据表各个字段的含义见表 8-11。

表 8-11　数据表 customer

字段名称	数据类型	相关说明
customer_id	smallint	主键(顾客 id)
store_id	tinyint	商店 id
first_name	varchar	顾客名字
last_name	varchar	顾客姓氏
email	varchar	顾客邮箱
address_id	smallint	地址 id
active	tinyint	活跃消费者
create_date	datetime	创建日期
last_update	timestamp	最后更新时间

12. 数据表 rental

数据表 rental 用于存储租借相关信息数据，该数据表各个字段的含义见表 8-12。

表 8-12 数据表 rental

字段名称	数据类型	相关说明
rental_id	int	主键(租借 id)
rental_date	datetime	租借日期
inventory_id	mediumint	库存编号
customer_id	smallint	客户 id
return_date	datetime	返还日期
staff_id	tinyint	员工 id
last_update	timestamp	最后更新时间

13. 数据表 payment

数据表 payment 用于存储租赁时付款的相关信息，该数据表各个字段的含义见表 8-13。

表 8-13 数据表 payment

字段名称	数据类型	相关说明
payment_id	smallint	主键(付款 id)
customer_id	smallint	顾客 id
staff_id	tinyint	员工 id
rental_id	int	租借 id
amount	decimal	数量
payment_date	datetime	付款日期
last_update	timestamp	最后更新时间

14. 数据表 country

数据表 country 用于存储国家 id 和对应的国家名称数据，该数据表各个字段的含义见表 8-14。

表 8-14 数据表 country

字段名称	数据类型	相关说明
country_id	smallint	主键(国家 id)
country	varchar	国家名称
last_update	timestamp	最后更新时间

15. 数据表 city

数据表 city 用于存储城市 id 和对应的城市名称以及所属国家 id 这一类数据，该数据表

各个字段的含义见表 8-15。

表 8-15 数据表 city

字段名称	数据类型	相关说明
city_id	smallint	主键(城市 id)
city	varchar	城市名称
country_id	smallint	国家 id
last_update	timestamp	最后更新时间

16. 数据表 address

数据表 address 用于存储城市地址及地址邮编、所属区域等相关信息，该数据表各个字段的含义见表 8-16。

表 8-16 数据表 address

字段名称	数据类型	相关说明
address_id	smallint	主键(地址 id)
address	varchar	地址名称
address2	varchar	地址名称 2
district	varchar	区域
city_id	smallint	城市 id
postal_code	varchar	邮编
phone	varchar	电话
last_update	timestamp	最后更新时间

8.3 案例实现

下面讲解如何构建 DVD 租赁商店数据仓库，以及使用 Kettle 工具实现抽取源数据库 sakila 中的数据，转换成符合 DVD 租赁业务的数据，并加载到 DVD 租赁商店数据仓库 sakila_dw 中。

8.3.1 构建 DVD 租赁商店数据仓库

我们基于数据库 sakila 构建一个星形模型的 DVD 租赁商店数据仓库，并命名为 sakila_dw。数据仓库 sakila_dw 中的事实表 fact_rental 是根据数据库 sakila 中的数据表 rental 创建的；维度表是根据数据表 sakila 中数据表的分类创建的，即从人员、时间、地点以及事件 4 个角度创建数据仓库 sakila_dw 的维度表，具体如下。

- 从人员角度创建维度表 dim_customer 和 dim_staff，分别表示租赁业务中的客户和员工。

- 从时间角度创建维度表 dim_date 和 dim_time，用于记录所有 DVD 的租赁时间和归还时间。
- 从地点角度创建维度表 dim_store，用于记录 DVD 光盘是从哪个商店租赁的。
- 从事件角度创建维度表 dim_actor 和 dim_film，其中 dim_actor 用于记录演员的基本信息，dim_film 用于记录电影的基本信息。由于电影是租赁和归还的实际对象，因此维度表 dim_film 应与事实表 fact_rental 关联。一部电影由多位演员出演，所以会有桥接表 dim_film_actor_bridge，该表将电影与演员相关联。

数据仓库 sakila_dw 中的维度表（dim_date 和 dim_time 除外）会对应数据库 sakila 中的某个数据表。例如，维度表 dim_store 对应数据表 store、维度表 dim_actor 对应数据表 actor。

本书为读者提供了创建 sakila_dw 数据仓库的 SQL 脚本文件，该脚本文件的名称为 sakila_dw_schema.sql，读者只使用 MySQL 图形化管理软件 SQLyog 运行 sakila_dw_schema.sql 脚本文件创建 sakila_dw 数据仓库即可。

8.3.2 加载日期数据至日期维度表

下面通过 Kettle 工具加载日期数据至日期维度表 dim_date，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_date，并添加“生成记录”控件、“增加序列”控件、“JavaScript 代码”控件、“表输出”控件以及 Hop 跳连接线，具体效果如图 8-11 所示。

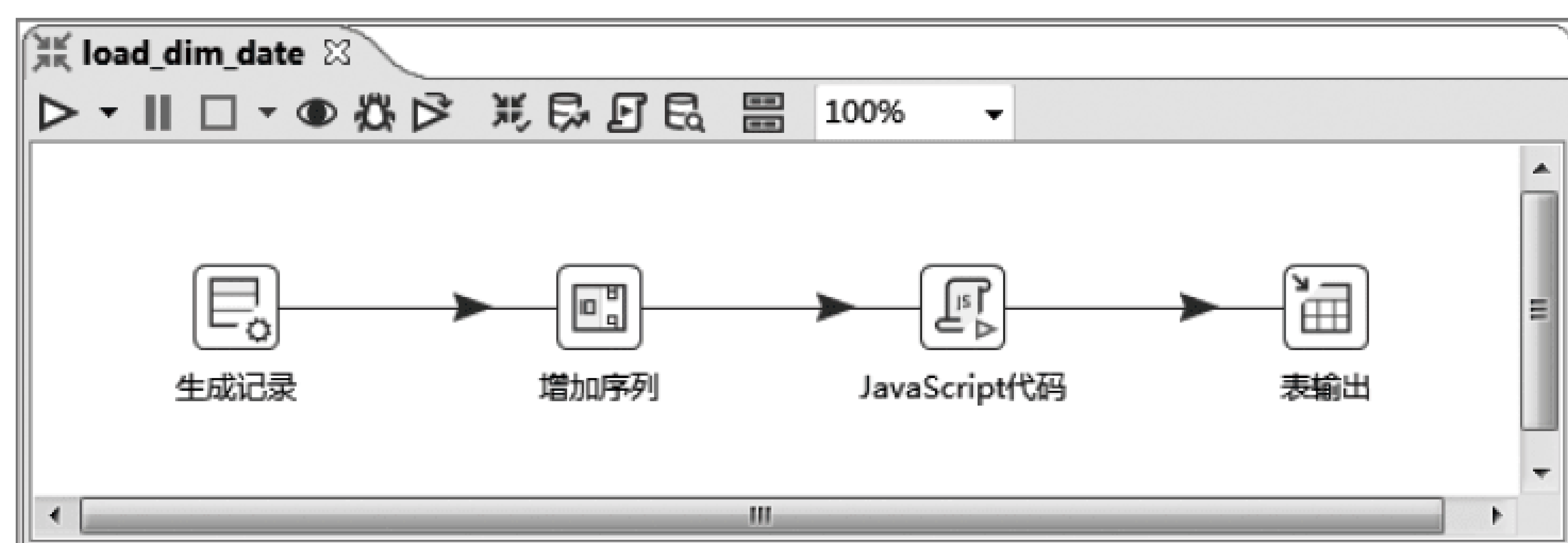


图 8-11 创建转换 load_dim_date

2. 配置“生成记录”控件

双击图 8-11 中的“生成记录”控件，进入“生成记录”界面，如图 8-12 所示。

在图 8-12 的“限制”处添加生成的日期，默认为 10，这里改为 3650，即生成 10 年的日期（10 * 365）；在“字段”框添加字段 language（语言）、country_code（国家码）、initial_date（初始化的日期），对生成的日期进行初始化，具体如图 8-13 所示。

在图 8-13 中单击“确定”按钮，完成“生成记录”控件的配置。

3. 配置“增加序列”控件

双击图 8-11 中的“增加序列”控件，进入“增加序列”界面，如图 8-14 所示。



图 8-12　“生成记录”界面



图 8-13　配置“生成记录”控件



图 8-14　“增加序列”界面

在图 8-14 中的“值的名称”处将 valuename 改为 DaySequence，即增加一列日期字段，用于改变“生成记录”控件生成的日期，具体如图 8-15 所示。



图 8-15 配置“增加序列”控件

在图 8-15 中单击“确定”按钮，完成“增加序列”控件的配置。

4. 配置“JavaScript 代码”控件

双击图 8-11 中的“JavaScript 代码”控件，进入“JavaScript 代码”界面，如图 8-16 所示。



图 8-16 “JavaScript 代码”界面

在图 8-16 中勾选“兼容模式?”复选框，使得“JavaScript 代码”控件的兼容性更强；在 Java Script 代码框中编写代码，具体代码如文件 8-1 所示。

文件 8-1 Script 1

```
1 //生成 locale、calendar
2 var locale =new java.util.Locale(language.getString(),country_code.getString());
3 var calendar =new java.util.GregorianCalendar(locale);
4 //设置时间、指定日历为当前日期
5 calendar.setTime(initial_date.getDate());
6 calendar.add(calendar.DAY_OF_MONTH,DaySequence.getInteger()-1);
7 //获取日期
8 var date =new java.util.Date(calendar.getTimeInMillis());
9 //生成短日期、中日期、长日期、全日期
10 var date_short =java.text.DateFormat.getDateInstance
11                               (java.text.DateFormat.SHORT,locale).format(date);
12 var date_medium =java.text.DateFormat.getDateInstance
13                               (java.text.DateFormat.MEDIUM,locale).format(date);
14 var date_long =java.text.DateFormat.getDateInstance
15                          (java.text.DateFormat.LONG,locale).format(date);
16 var date_full =java.text.DateFormat.getDateInstance
17                          (java.text.DateFormat.FULL,locale).format(date);
18 //简单格式化
19 var simpleDateFormat =java.text.SimpleDateFormat("D",locale);
20 //某天在年的第几天、某天在月的第几天
21 var day_in_year =simpleDateFormat.format(date);
22 simpleDateFormat.applyPattern("d");
23 var day_in_month =simpleDateFormat.format(date);
24 simpleDateFormat.applyPattern("EEEE");
25 //星期的名称、星期的缩写
26 var day_name =simpleDateFormat.format(date);
27 simpleDateFormat.applyPattern("E");
28 var day_abbreviation =simpleDateFormat.format(date);
29 simpleDateFormat.applyPattern("ww");
30 //一年的第几周、一月的第几周
31 var week_in_year =simpleDateFormat.format(date);
32 simpleDateFormat.applyPattern("W");
33 var week_in_month =simpleDateFormat.format(date);
34 simpleDateFormat.applyPattern("MM");
35 //设置月份,即月份的名称、月份的缩写
36 var month_number =simpleDateFormat.format(date);
37 simpleDateFormat.applyPattern("MMMM");
38 var month_name =simpleDateFormat.format(date);
39 simpleDateFormat.applyPattern("MMM");
40 var month_abbreviation =simpleDateFormat.format(date);
41 simpleDateFormat.applyPattern("yy");
42 //设置年份,即年的格式为两位、四位
43 var year2 =simpleDateFormat.format(date);
44 simpleDateFormat.applyPattern("yyyy");
45 var year4 =simpleDateFormat.format(date);
46 //设置季度
47 var quarter_name ="Q";
48 var quarter_number;
```

```
49 switch(parseInt(month_number)){
50         case 1:case 2:case 3:quarter_number ="1";break;
51         case 4:case 5:case 6:quarter_number ="2";break;
52         case 7:case 8:case 9:quarter_number ="3";break;
53         case 10:case 11:case 12:quarter_number ="4";break;
54 }
55 quarter_name +=quarter_number;
56 //定义常量
57 var yes ="yes";
58 var no ="no";
59 //获取周的第一天
60 var first_day_of_week =calendar.getFirstDayOfWeek();
61 var day_of_week =java.util.Calendar.DAY_OF_WEEK;
62 //判断是否为周的第一天
63 var is_first_day_in_week;
64 if(first_day_of_week ==calendar.get(day_of_week)){
65     is_first_day_in_week =yes;
66 }else{
67     is_first_day_in_week =no;
68 }
69 //获取日历的下一天
70 calendar.add(calendar.DAY_OF_MONTH,1);
71 //获取下一天
72 var next_day =new java.util.Date(calendar.getTimeInMillis());
73 //判断是否为周的最后一天、是否为月的第一天、是否为月的最后一天
74 var is_last_day_in_week;
75 if(first_day_of_week ==calendar.get(day_of_week)){
76     is_last_day_in_week =yes;
77 }else{
78     is_last_day_in_week =no;
79 }
80 var is_first_day_in_month;
81 if(day_in_month ==1){
82     is_first_day_in_month =yes;
83 }else{
84     is_first_day_in_month =no;
85 }
86 var is_last_day_in_month;
87 if(java.text.SimpleDateFormat("d",locale).format(next_day)==1){
88     is_last_day_in_month =yes;
89 }else{
90     is_last_day_in_month =no;
91 }
92 //设置年_季度、年_月份、年_月的缩写
93 var year_quarter =year4 +"-" +quarter_name;
94 var year_month_number =year4 +"-" +month_number;
95 var year_month_abbreviation =year4 +"-" +month_abbreviation;
96 //生成日期代理键(唯一键)
97 var date_key =year4 +month_number +(day_in_month<10?"0":"") +day_in_month;
```

上述代码中，第 1～19 行代码创建 locale、calendar 以及 date 对象，设置时间和日期；第 20～55 行代码对天数、星期、月份、年份、季度进行格式化；第 56～68 行代码获取周的第一天；第 69～91 行代码获取日历的下一天，判断是否为周的最后一天、是否为月的第一天、是否为月的最后一天；第 92～95 行代码设置年_季度、年_月份、年_月的缩写；第 97 行代码生成日期代理键。

单击图 8-16 中的“获取变量”按钮，将代码中定义的变量添加至字段框中，具体如图 8-17 所示。

字段

#	字段名称	改名为	类型	长度	精度	替换 'Fieldname' 或 'Rename to'值
1	date		Date			否
2	date_short		String			否
3	date_medium		String			否
4	date_long		String			否
5	date_full		String			否
6	day_in_year		String			否
7	day_in_month		String			否
8	day_name		String			否
9	day_abbreviation		String			否
10	week_in_year		String			否
11	week_in_month		String			否
12	month_number		String			否
13	month_name		String			否
14	month_abbreviation		String			否
15	year2		String			否
16	year4		String			否
17	quarter_name		String			否
18	quarter_number		String			否
19	is_first_day_in_week		String			否
20	is_last_day_in_week		String			否
21	is_first_day_in_month		String			否
22	is_last_day_in_month		String			否
23	year_quarter		String			否
24	year_month_number		String			否
25	year_month_abbreviation		String			否
26	date_key		String			否

图 8-17　添加变量至字段框中

“JavaScript 代码”控件的配置效果如图 8-18 所示，单击“确定”按钮，完成“JavaScript 代码”控件的配置。

5. 配置“表输出”控件

双击图 8-11 中的“表输出”控件，进入“表输出”界面，如图 8-19 所示。

在图 8-19 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-20 所示。

在图 8-19 中单击目标表右侧的“浏览”按钮，选择输出的目标表，即维度表 dim_date；勾选“指定数据库字段”复选框，用于将维度表字段与 JavaScript 控件流中的变量字段进行匹配，具体如图 8-21 所示。

在图 8-21 中单击“数据库字段”选项卡，弹出“数据库字段”选项卡界面，在该界面下单击“输入字段映射”按钮，弹出“映射匹配”对话框，如图 8-22 所示。

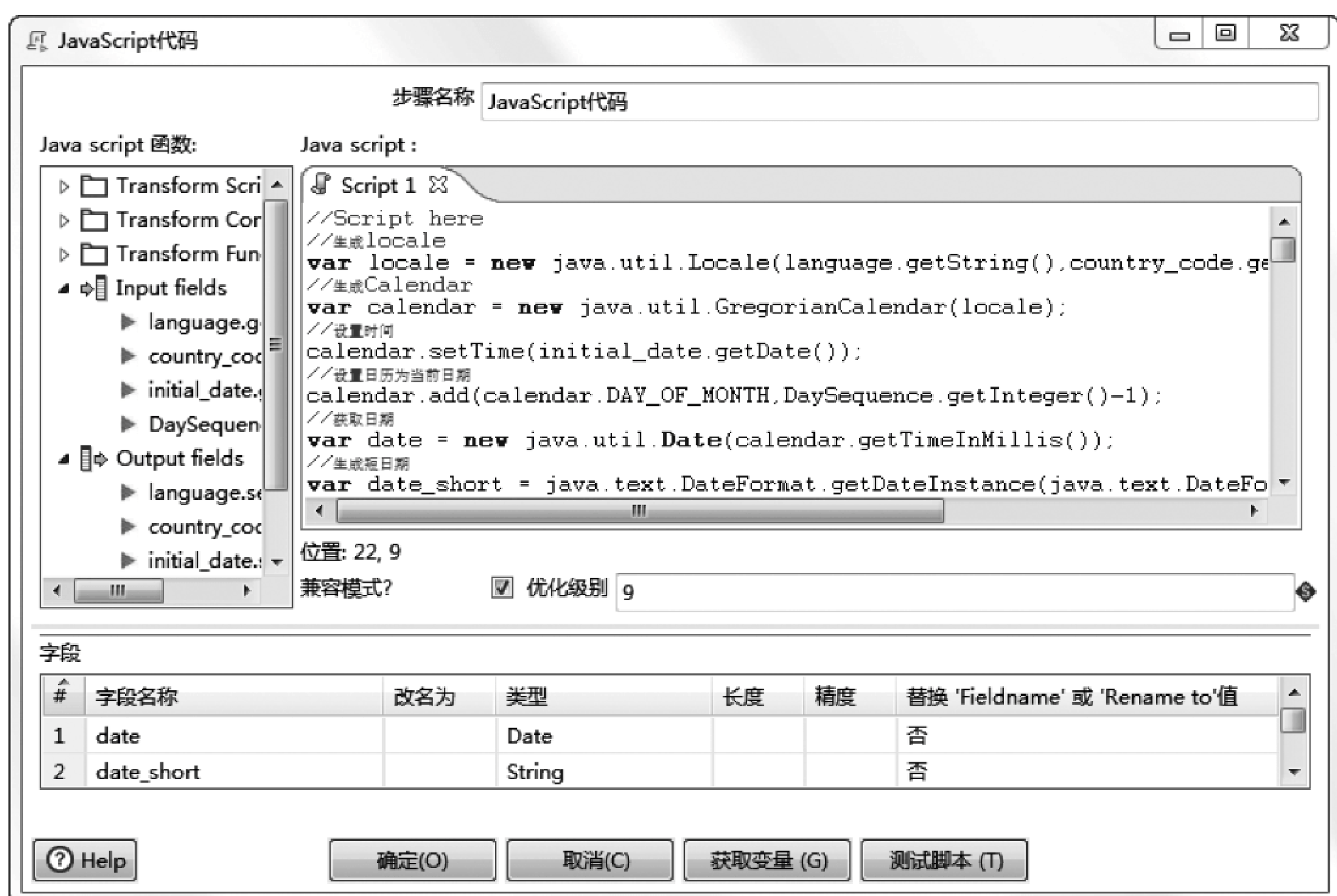


图 8-18 “JavaScript 代码”控件的配置



图 8-19 “表输出”界面

在图 8-22 中依次选中“源字段”中的字段和“目标字段”中的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 8-23 所示。

在图 8-23 中单击“映射匹配”对话框中的“确定”按钮，“表输出”控件配置的效果图如图 8-24 所示，单击“确定”按钮，完成“表输出”控件的配置。



图 8-20 MySQL 数据库连接的配置



图 8-21 指定输出的目标表

6. 运行转换 load_dim_date

单击转换工作区顶部的▷按钮,运行创建的转换 load_dim_date,实现加载日期数据至日期维度表 dim_date 中,具体如图 8-25 所示。

从图 8-25 中执行结果的“步骤度量”可以看出,“生成记录”控件写入 3650 条数据;“增

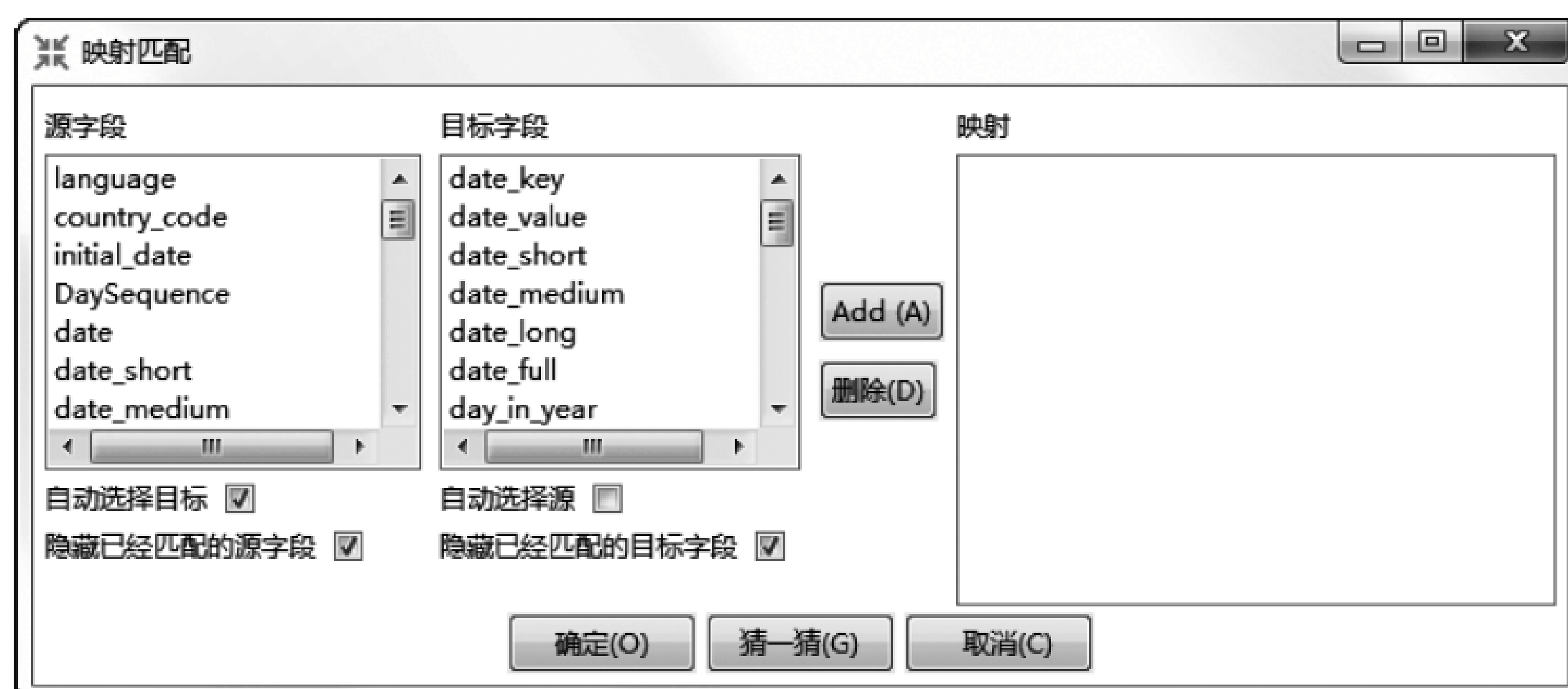


图 8-22 “映射匹配”对话框

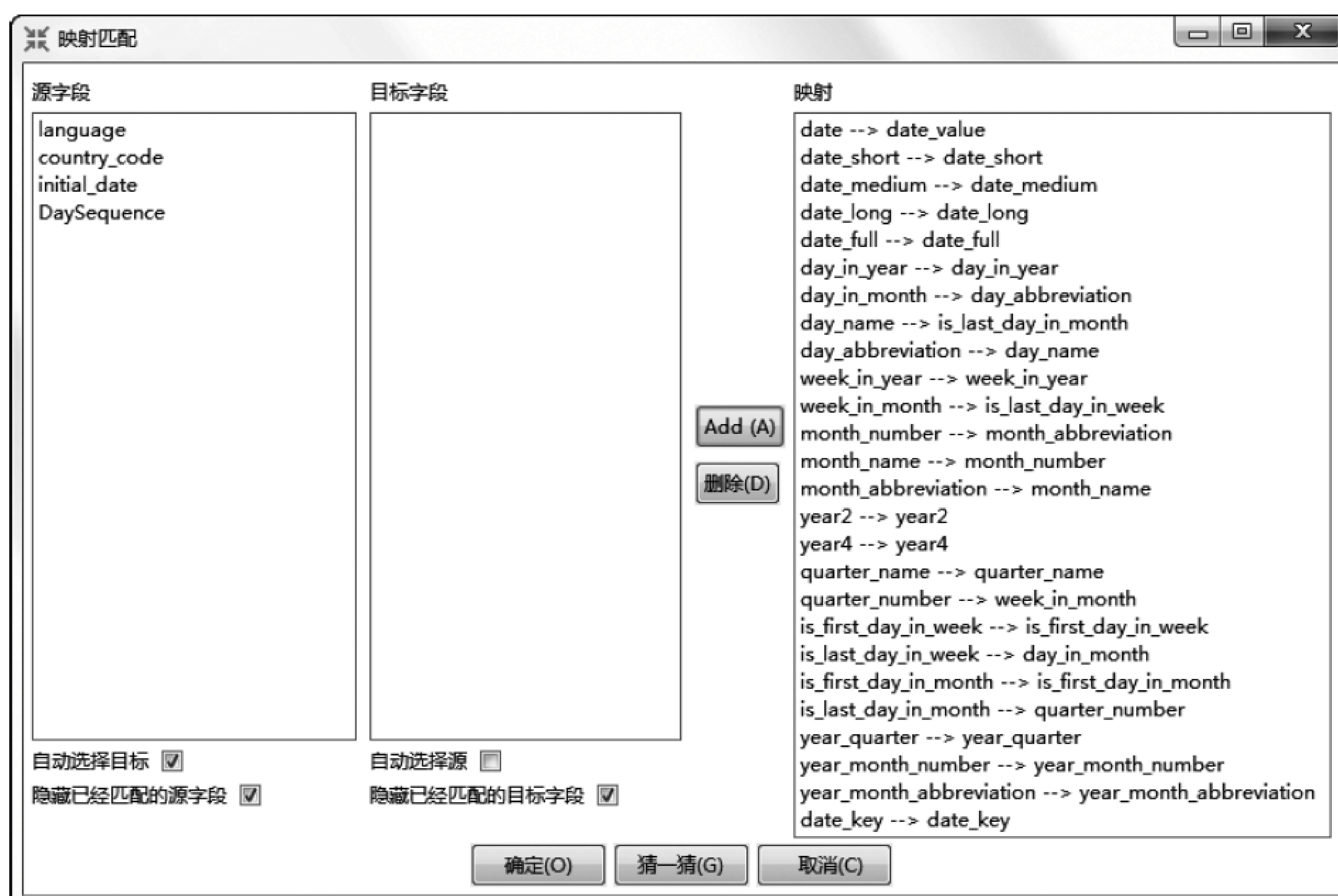


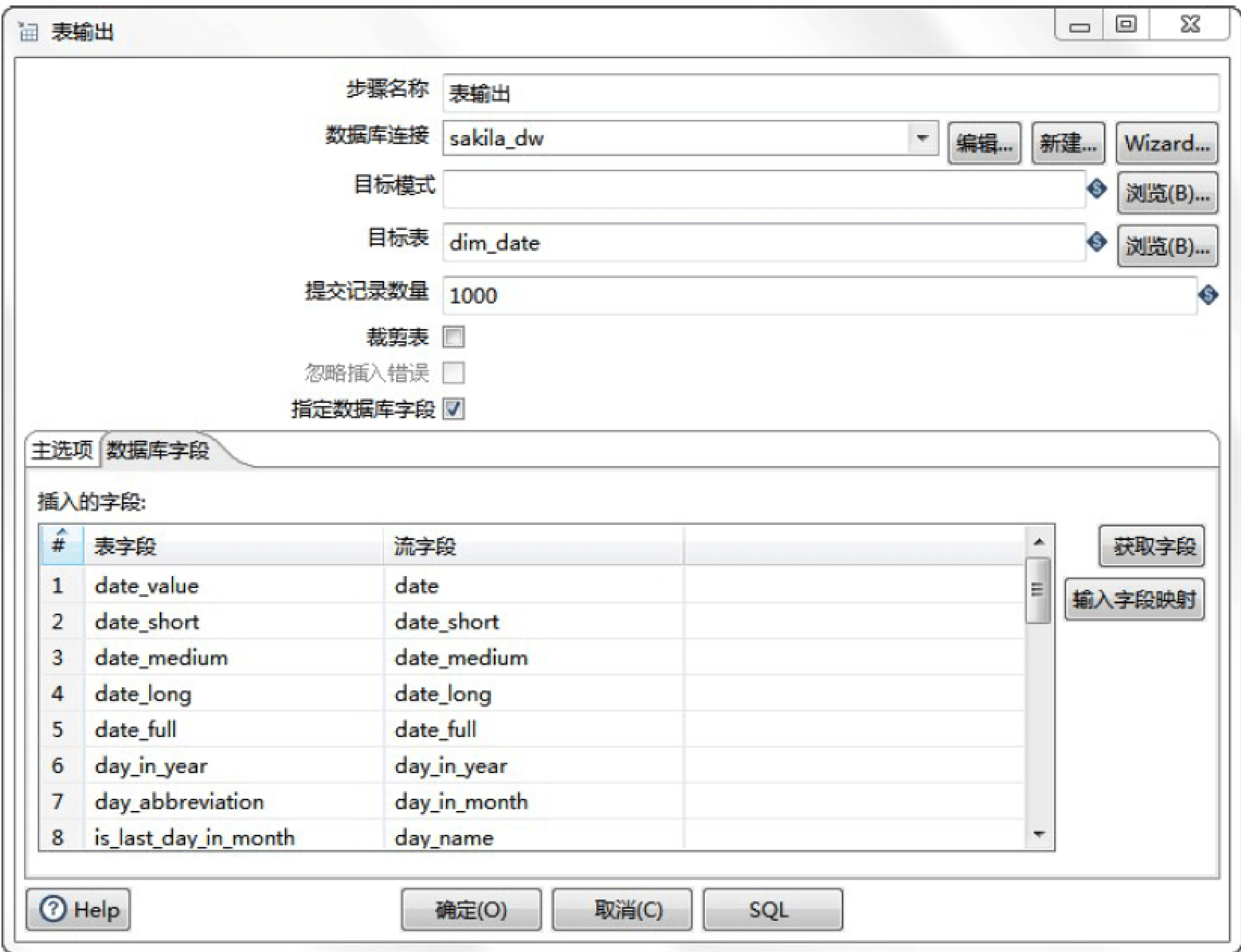
图 8-23 设置映射匹配

加序列”控件从“生成记录”控件中读取 3650 条数据并写入该控件中；“JavaScript 代码”控件从“增加序列”控件中读取 3650 条数据并写入该控件；“表输出”控件从“Java Script 代码”控件中读取 3650 条数据并写入该控件，最终进行输出。

7. 查看维度表 dim_date 中的数据

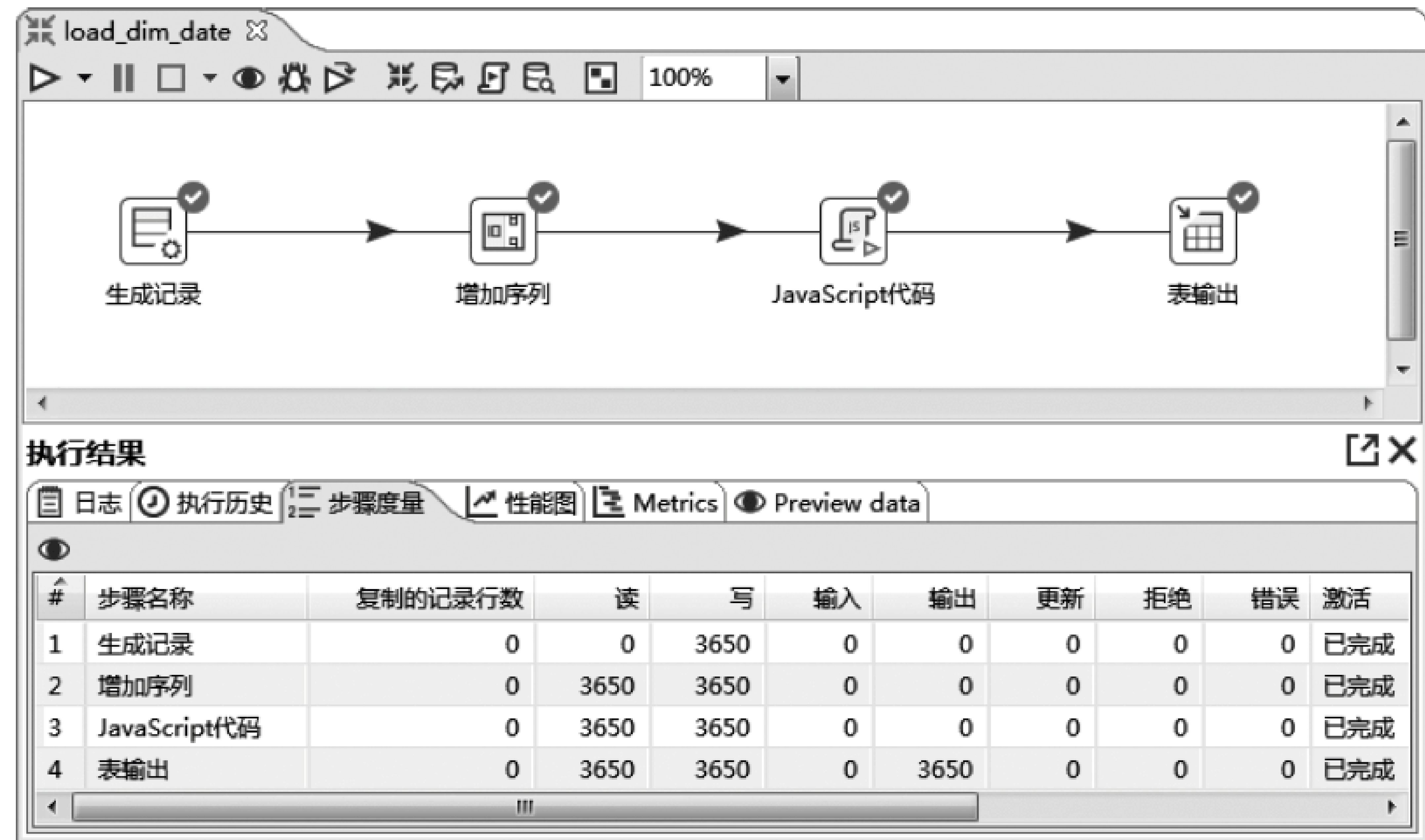
通过 SQLyog 工具，查看维度表 dim_date 是否已成功插入 3650 条日期数据，查看结果如图 8-26 所示（这里只截取了部分数据）。

从图 8-26 中可以看出，维度表 dim_date 中已插入数据，说明我们成功实现了加载日期数据至日期维度表 dim_date 中。



#	表字段	流字段
1	date_value	date
2	date_short	date_short
3	date_medium	date_medium
4	date_long	date_long
5	date_full	date_full
6	day_in_year	day_in_year
7	day_abbreviation	day_in_month
8	is_last_day_in_month	day_name

图 8-24　"表输出"控件配置的效果图



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	生成记录	0	0	3650	0	0	0	0	0	已完成
2	增加序列	0	3650	3650	0	0	0	0	0	已完成
3	JavaScript代码	0	3650	3650	0	0	0	0	0	已完成
4	表输出	0	3650	3650	0	3650	0	0	0	已完成

图 8-25　运行转换 load_dim_date

date_key	date_v...	date_short	date_medium	date_long	date_full	day_in_year
20091228	2009-12-28	12/28/09	Dec 28, 2009	December 28, 2009	Monday, December 28, 2009	362
20091227	2009-12-27	12/27/09	Dec 27, 2009	December 27, 2009	Sunday, December 27, 2009	361
20091226	2009-12-26	12/26/09	Dec 26, 2009	December 26, 2009	Saturday, December 26, 2009	360
20091225	2009-12-25	12/25/09	Dec 25, 2009	December 25, 2009	Friday, December 25, 2009	359
20091224	2009-12-24	12/24/09	Dec 24, 2009	December 24, 2009	Thursday, December 24, 2009	358
20091223	2009-12-23	12/23/09	Dec 23, 2009	December 23, 2009	Wednesday, December 23, 2009	357
20091222	2009-12-22	12/22/09	Dec 22, 2009	December 22, 2009	Tuesday, December 22, 2009	356
20091221	2009-12-21	12/21/09	Dec 21, 2009	December 21, 2009	Monday, December 21, 2009	355
20091220	2009-12-20	12/20/09	Dec 20, 2009	December 20, 2009	Sunday, December 20, 2009	354

图 8-26　维度表 dim_date

8.3.3 加载时间数据至时间维度表

下面通过 Kettle 工具加载时间数据至时间维度表 dim_time，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_time，并添加“生成记录”控件、“增加序列”控件、“JavaScript 代码”控件、“记录关联（笛卡儿输出）”控件（该组件在整个转换中不做任何配置，因此后续步骤不做讲解）、“表输出”控件以及 Hop 跳连接线，具体效果如图 8-27 所示。

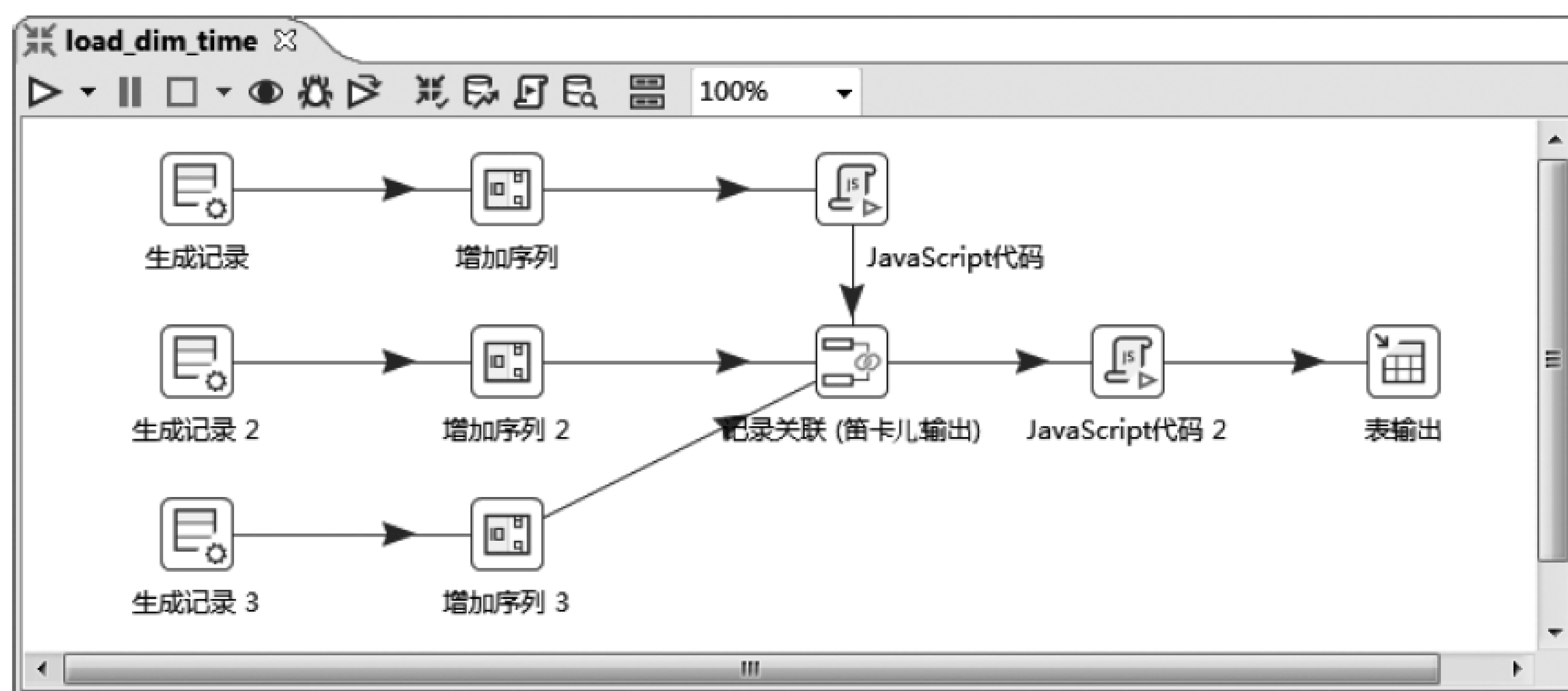


图 8-27 创建转换 load_dim_time

2. 配置“生成记录”控件

双击图 8-27 中的“生成记录”控件，进入“生成记录”界面，在“限制”后的文本框添加 24 表示生成 24 条数据（24 小时从 0 点开始至 23 点结束共 24 条数据）；在“字段”框添加生成字段的名称、字段类型及默认值为 0，具体如图 8-28 所示。

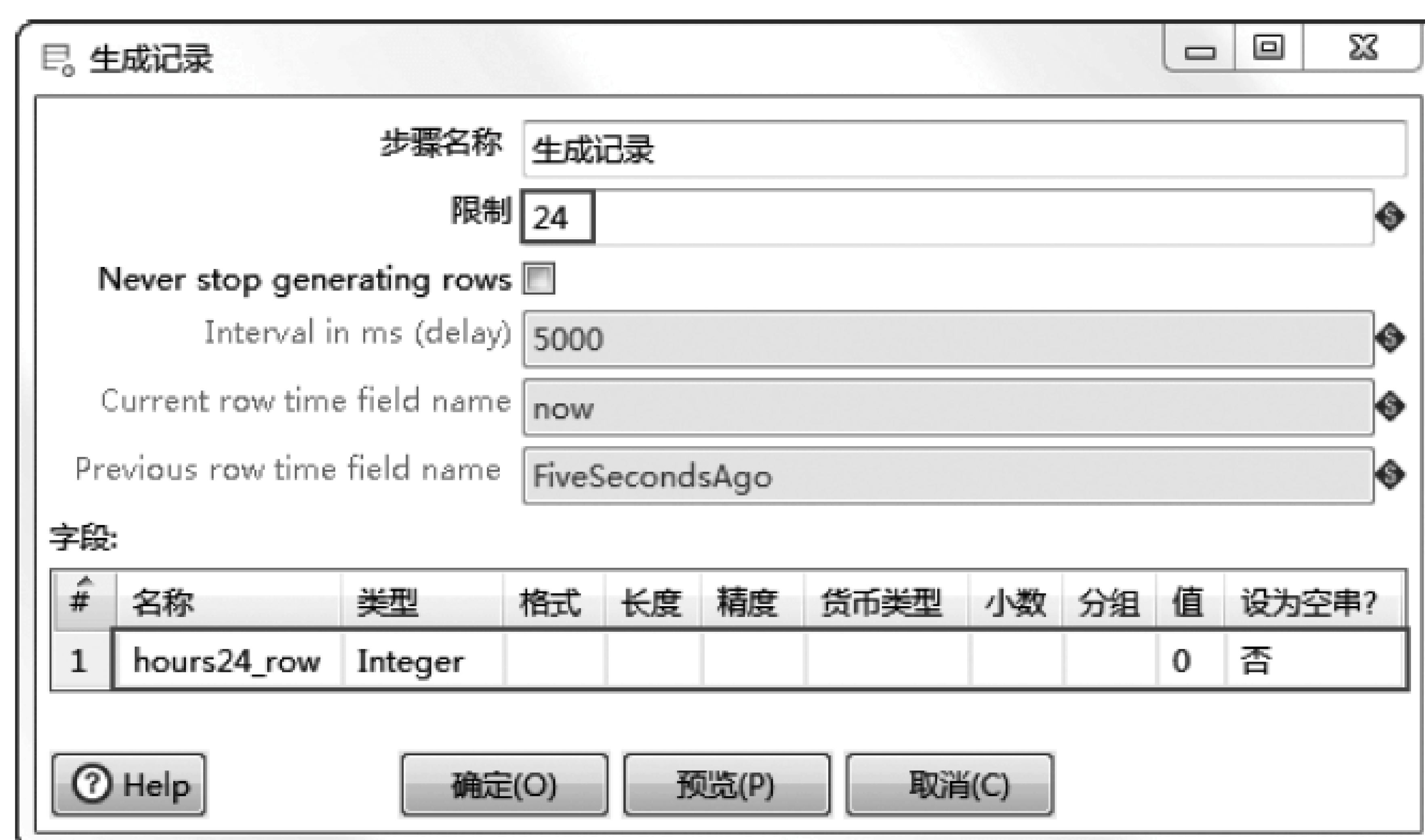


图 8-28 配置“生成记录”控件

在图 8-28 中单击“确定”按钮，完成“生成记录”控件的配置。

3. 配置“增加序列”控件

双击图 8-27 中的“增加序列”控件，进入“增加序列”界面，在“值的名称”处将 valuename 改为 hours24，即增加一列小时字段，在起始值后的文本框内将默认值 1 修改为 0，代表从 0 开始生成 24 条数据（上一步“生成记录”控件限制了条数），即生成数据为 0～23。由于时间由时分秒构成，因此需要生成时分秒字段的数据，这里先生成 24 小时数据，后续步骤中会生成 60 分和 60 秒的数据，具体如图 8-29 所示。

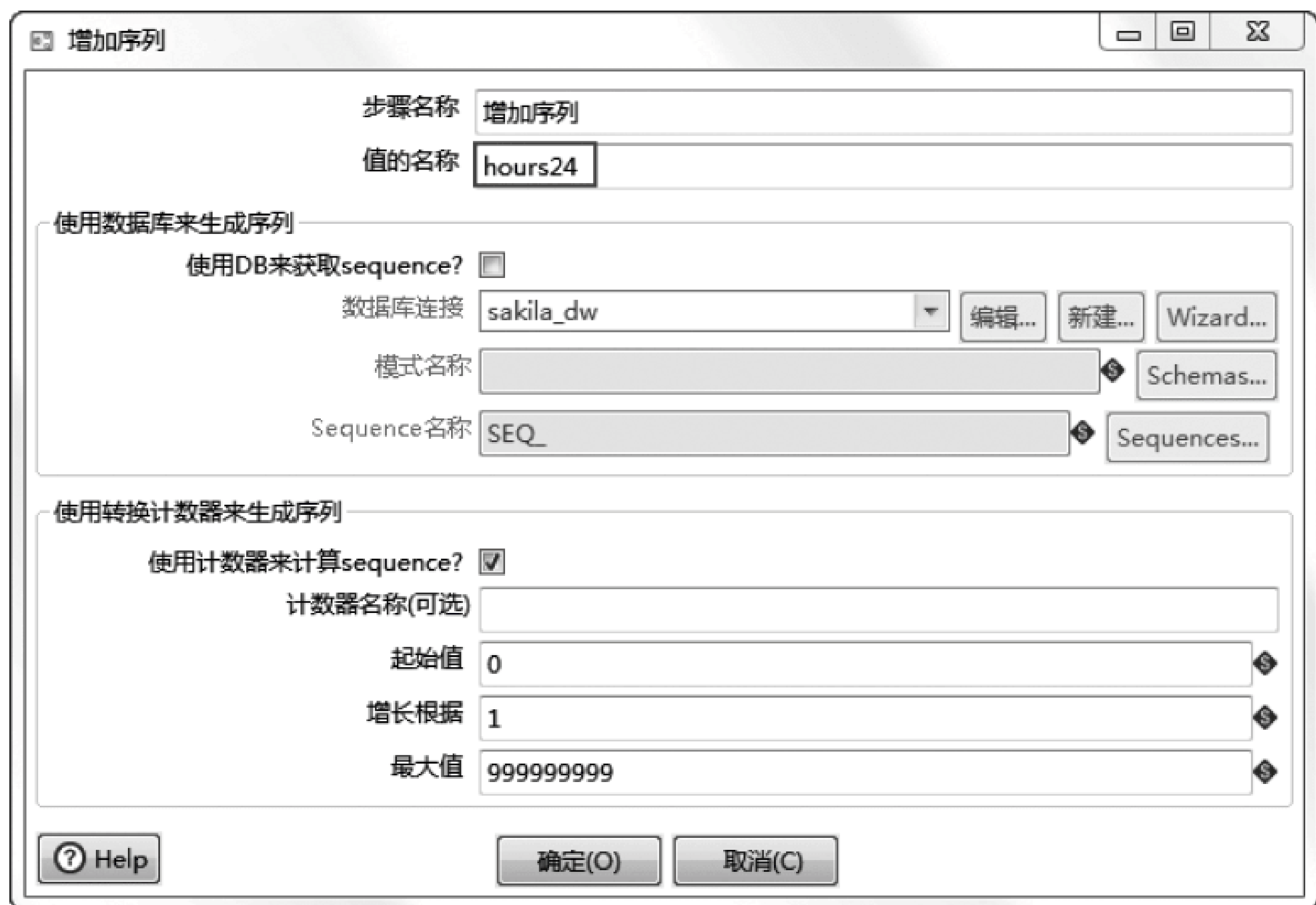


图 8-29　配置“增加序列”控件

在图 8-29 中单击“确定”按钮，完成“增加序列”控件的配置。

4. 配置“JavaScript 代码”控件

双击图 8-27 中的“JavaScript 代码”控件，进入“JavaScript 代码”界面，勾选“兼容模式?”复选框，使得“JavaScript 代码”控件的兼容性更强；在 JavaScript 代码框中编写代码；单击“获取变量”按钮，将代码中定义的变量添加至字段框，具体如图 8-30 所示。

在图 8-30 中单击“确定”按钮，完成“JavaScript 代码”控件的配置。

5. 配置“生成记录 2”控件

双击图 8-27 中的“生成记录 2”控件，进入“生成记录”界面，在“限制”后的文本框添加 60 表示生成 60 条数据（60 分钟从 0 分开始至 59 分结束共 60 条数据）；在“字段”框添加生成字段的名称、字段类型及默认值为 0，具体如图 8-31 所示。

在图 8-31 中单击“确定”按钮，完成“生成记录 2”控件的配置。

6. 配置“增加序列 2”控件

双击图 8-27 中的“增加序列 2”控件，进入“增加序列”界面，将“值的名称”处的



图 8-30 配置"JavaScript 代码"控件

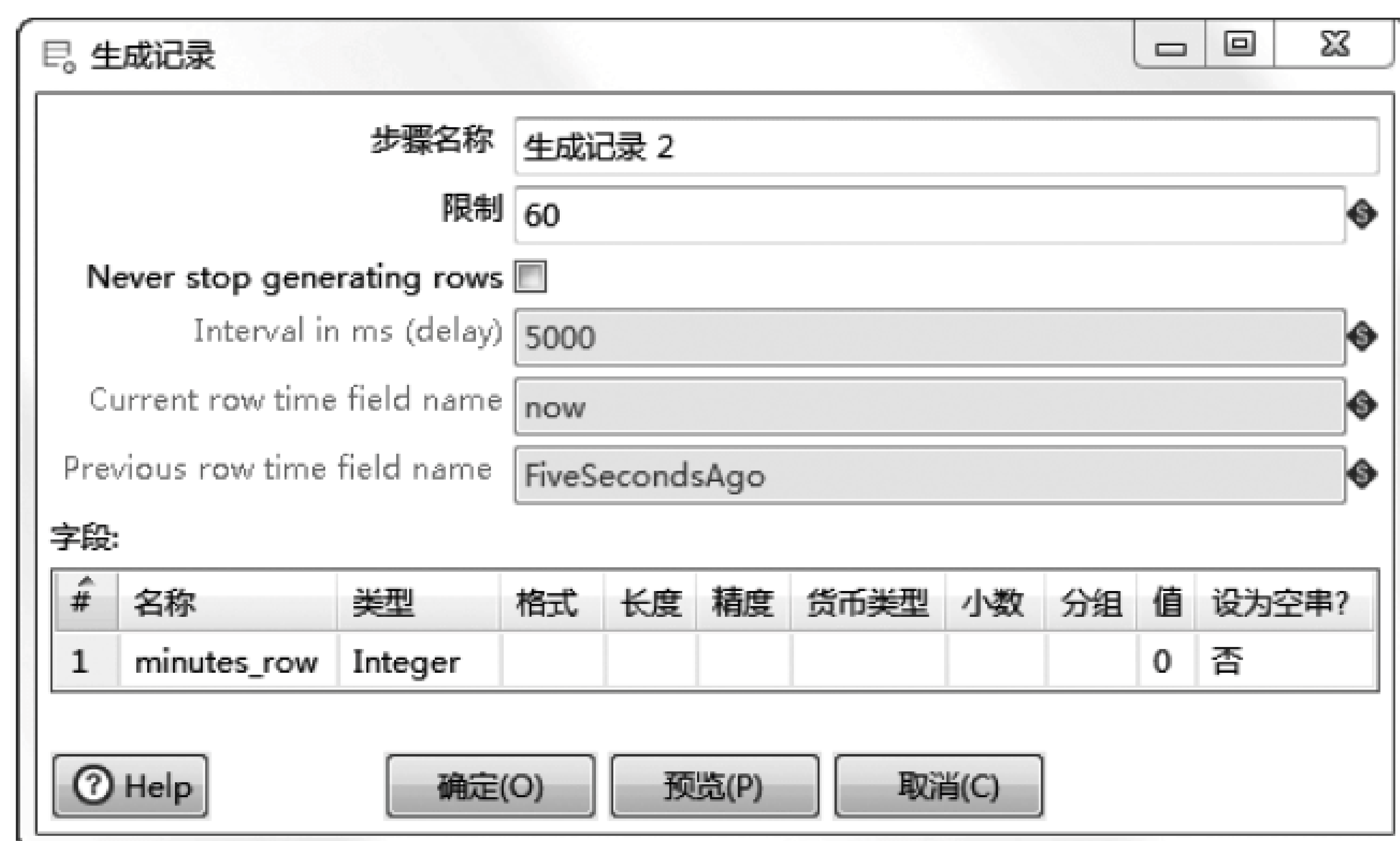


图 8-31 配置"生成记录"控件

valuename 改为 minutes，即增加一列分钟字段，用于记录分钟数，在起始值后的文本框内将默认值 1 修改为 0，代表从 0 开始生成 60 条数据(上一步"生成记录 2"控件限制了条数)，即生成数据为 0～59。具体如图 8-32 所示。

在图 8-32 中单击"确定"按钮，完成"增加序列 2"控件的配置。

7. 配置"生成记录 3"控件

双击图 8-27 中的"生成记录 3"控件，进入"生成记录"界面，在"限制"后的文本框添加 60 表示生成 60 条数据(60 秒钟从 0 秒开始至 59 秒结束共 60 条数据)；在"字段"框添加生成字段的名称、字段类型及默认值为 0，具体如图 8-33 所示。



图 8-32　配置“增加序列 2”控件

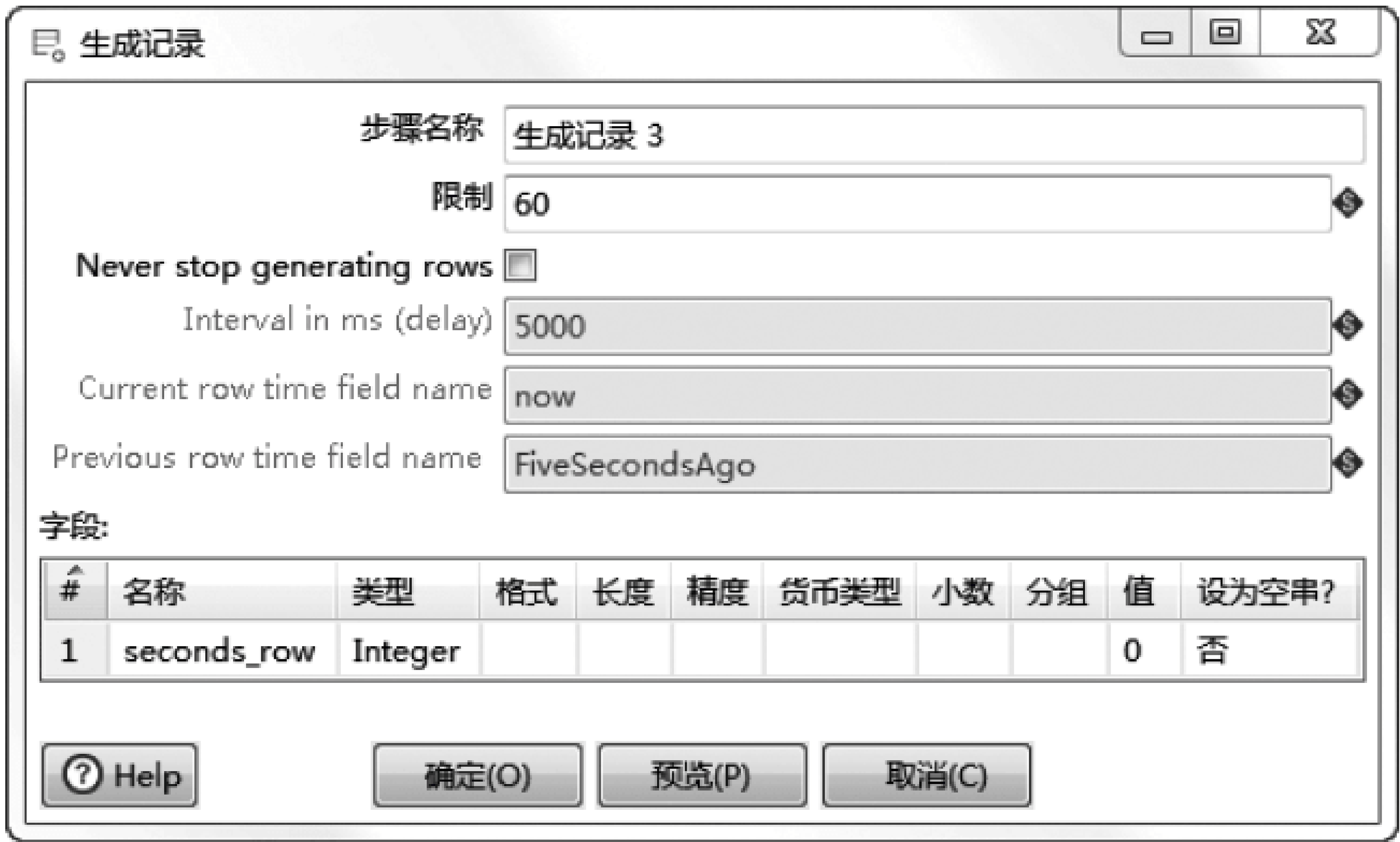


图 8-33　配置“生成记录”控件

在图 8-33 中单击“确定”按钮，完成“生成记录”控件的配置。

8. 配置“增加序列 3”控件

双击图 8-27 中的“增加序列 3”控件，进入“增加序列”界面，将“值的名称”处的 valuename 改为 seconds，即增加一列秒字段，用于记录秒数，在起始值后的文本框内将默认值 1 修改为 0，代表从 0 开始生成 60 条数据(上一步“生成记录 3”控件限制了条数)，即生成数据为 0～59。具体如图 8-34 所示。

在图 8-34 中单击“确定”按钮，完成“增加序列 3”控件的配置。

9. 配置“JavaScript 代码 2”控件

双击图 8-27 中的“JavaScript 代码 2”控件，进入“JavaScript 代码”界面，勾选“兼容模式?”复选框，使得“JavaScript 代码 2”控件的兼容性更强；在 JavaScript 代码框中编写代码；



图 8-34 配置“增加序列 3”控件

单击“获取变量”按钮,将代码中定义的变量添加至字段框,具体如图 8-35 所示。



图 8-35 配置“JavaScript 代码 2”控件

在图 8-35 中单击“确定”按钮,完成“JavaScript 代码 2”控件的配置。

10. 配置“表输出”控件

双击图 8-27 中的“表输出”控件,进入“表输出”界面,如图 8-36 所示。

在图 8-36 中单击“新建”按钮,配置数据库连接,配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-37 所示。



图 8-36　“表输出”界面



图 8-37　MySQL 数据库连接的配置

在图 8-36 中单击目标表右侧的“浏览”按钮，选择输出的目标表，即维度表 dim_time；勾选“指定数据库字段”复选框，用于将维度表 dim_time 的字段与 JavaScript 控件流中的字段进行匹配，具体如图 8-38 所示。

在图 8-38 中单击“数据库字段”选项卡，切换到“数据库字段”选项卡界面，并在该界面下单击“输入字段映射”按钮，弹出“映射匹配”对话框，如图 8-39 所示。

在图 8-39 中依次选中“源字段”中的字段和“目标字段”中的字段，再单击 Add 按钮，将一对映射字段添加至“映射”框中，若“源字段”中的字段和“目标字段”中的字段相同，则可以单击“猜一猜”按钮，让 Kettle 自动实现映射，具体如图 8-40 所示。

在图 8-40 中单击“映射匹配”对话框中的“确定”按钮，“表输出”控件配置的效果图如图 8-41 所示，单击“确定”按钮，完成“表输出”控件的配置。



图 8-38 指定输出的目标表

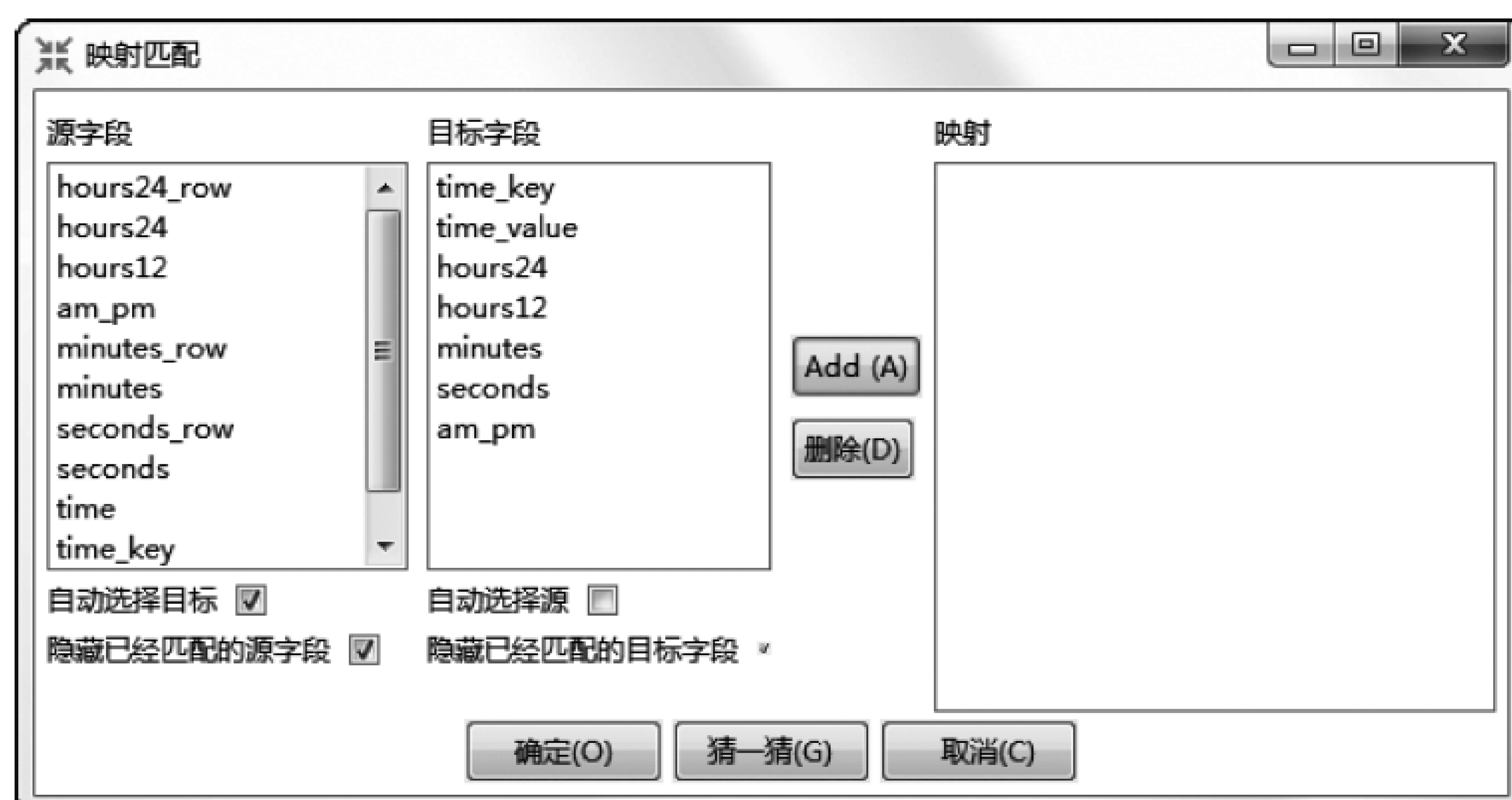


图 8-39 “映射匹配”对话框

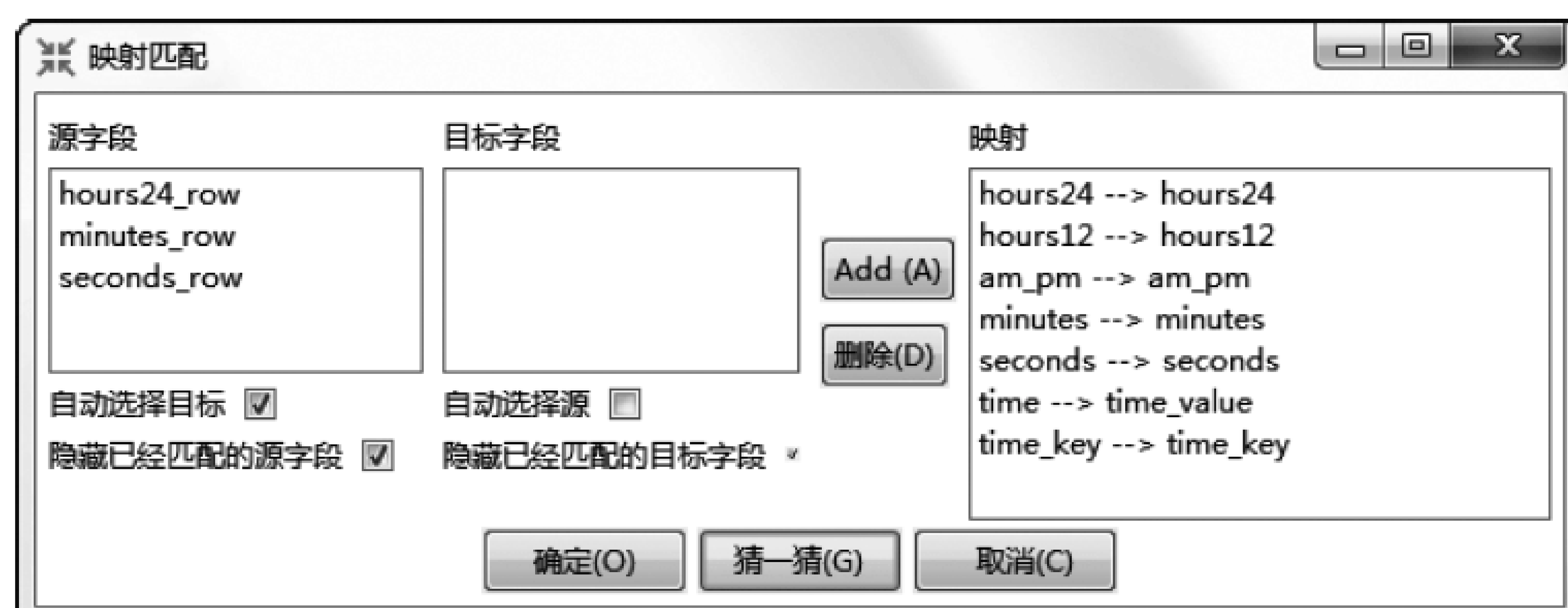


图 8-40 设置映射匹配

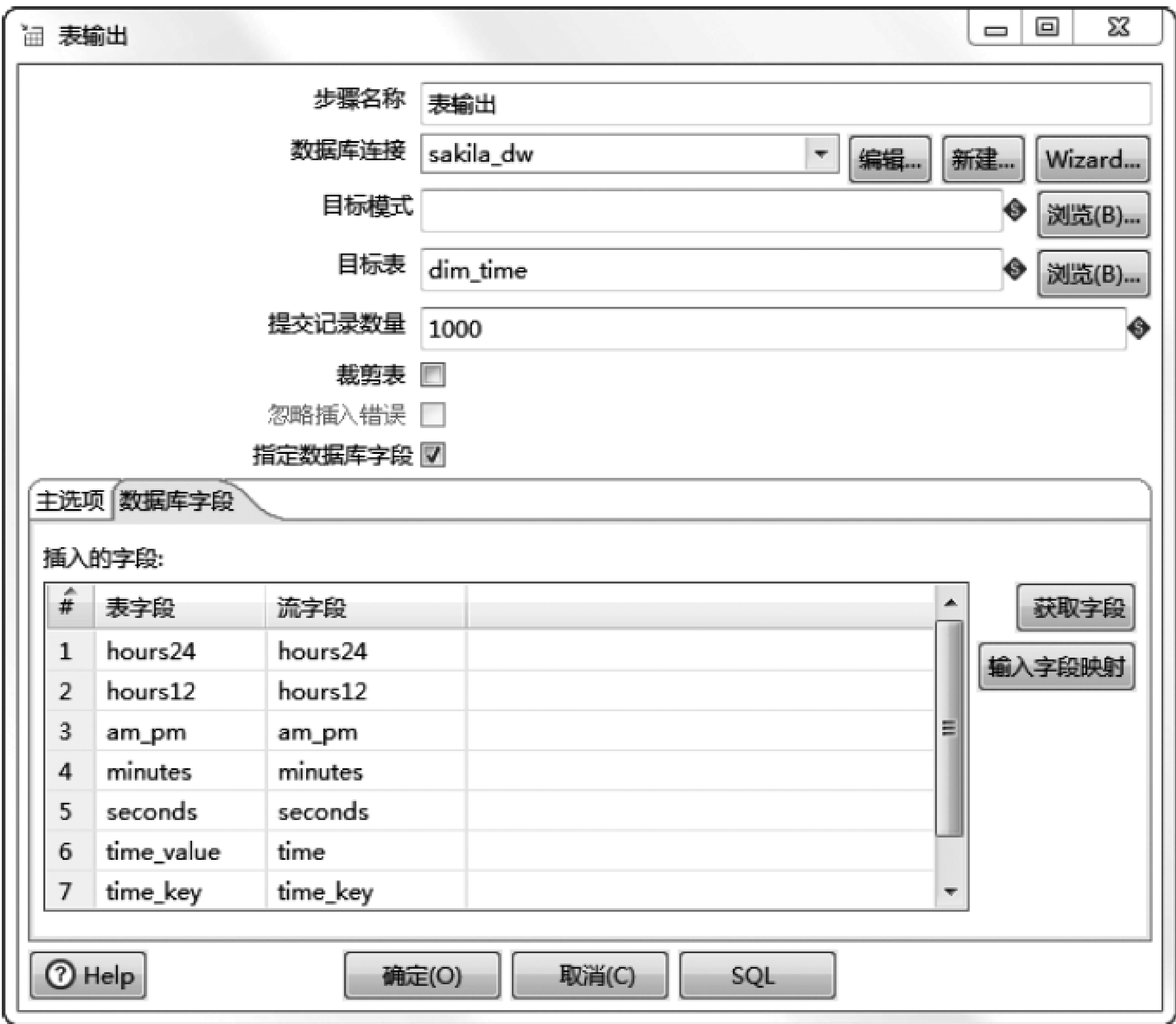
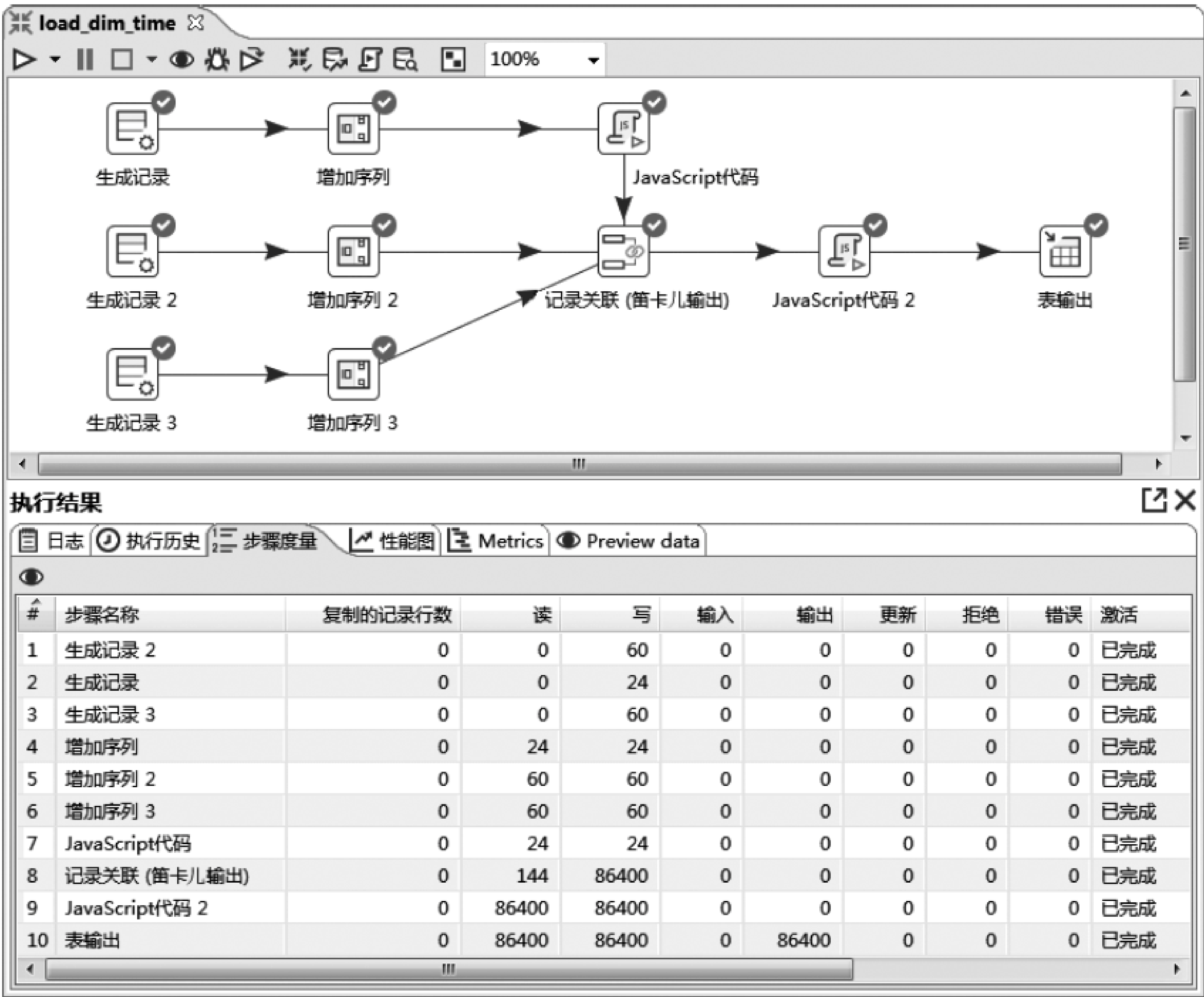


图 8-41　“表输出”控件配置的效果图

11. 运行转换 load_dim_time

单击转换工作区顶部的▷按钮，运行创建的转换 load_dim_date，实现加载时间数据至时间维度表 dim_time 中，具体如图 8-42 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	生成记录 2	0	0	60	0	0	0	0	0	已完成
2	生成记录	0	0	24	0	0	0	0	0	已完成
3	生成记录 3	0	0	60	0	0	0	0	0	已完成
4	增加序列	0	24	24	0	0	0	0	0	已完成
5	增加序列 2	0	60	60	0	0	0	0	0	已完成
6	增加序列 3	0	60	60	0	0	0	0	0	已完成
7	JavaScript代码	0	24	24	0	0	0	0	0	已完成
8	记录关联 (笛卡儿输出)	0	144	86400	0	0	0	0	0	已完成
9	JavaScript代码 2	0	86400	86400	0	0	0	0	0	已完成
10	表输出	0	86400	86400	0	86400	0	0	0	已完成

图 8-42　运行转换 load_dim_time

从图 8-42 中执行结果的“步骤度量”可以看出，“生成记录 2”控件写入 60 条数据(分钟数据)；“生成记录”控件写入 24 条数据(小时数据)；“生成记录 3”控件写入 60 条数据(秒数据)；“增加序列”控件从“生成记录”控件中读取 24 条数据并写入该控件；“增加序列 2”控件从“生成记录 2”控件中读取 60 条数据并写入该控件；“增加序列 3”控件从“生成记录 3”控件中读取 60 条数据并写入该控件；“JavaScript 代码”控件从“增加序列”控件中读取 24 条数据并写入该控件；“记录关联(笛卡儿输出)”控件将 Java Script 控件、“增加序列 2”控件、“增加序列 3”控件进行记录关联，读取到 144 条数据并写入该控件 86400 条数据；“JavaScript 代码 2”控件从“记录关联”控件中读取 86400 条数据并写入该控件；“表输出”控件从 Java Script 2 控件中读取 86400 条数据并写入该控件，最终进行输出。

12. 查看维度表 dim_time 中的数据

通过 SQLyog 工具，查看维度表 dim_time 是否已成功插入 86400 条时间数据，查看结果如图 8-43 所示(这里只截取了部分数据)。

time_key	time_value	hours24	hours12	minutes	seconds	am_pm
0	00:00:00	0	0	0	0	AM
1	00:00:01	0	0	0	1	AM
2	00:00:02	0	0	0	2	AM
3	00:00:03	0	0	0	3	AM
4	00:00:04	0	0	0	4	AM
5	00:00:05	0	0	0	5	AM
6	00:00:06	0	0	0	6	AM
7	00:00:07	0	0	0	7	AM

图 8-43 维度表 dim_time

从图 8-43 中可以看出，维度表 dim_time 中已插入数据，说明我们成功实现了加载时间数据至时间维度表 dim_time 中。

8.3.4 加载员工数据至员工维度表

下面通过 Kettle 工具加载员工数据至员工维度表 dim_staff，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_staff，并添加“表输入”控件、“字段选择”控件、“值映射”控件、“维度查询/更新”控件以及 Hop 跳连接线，具体如图 8-44 所示。

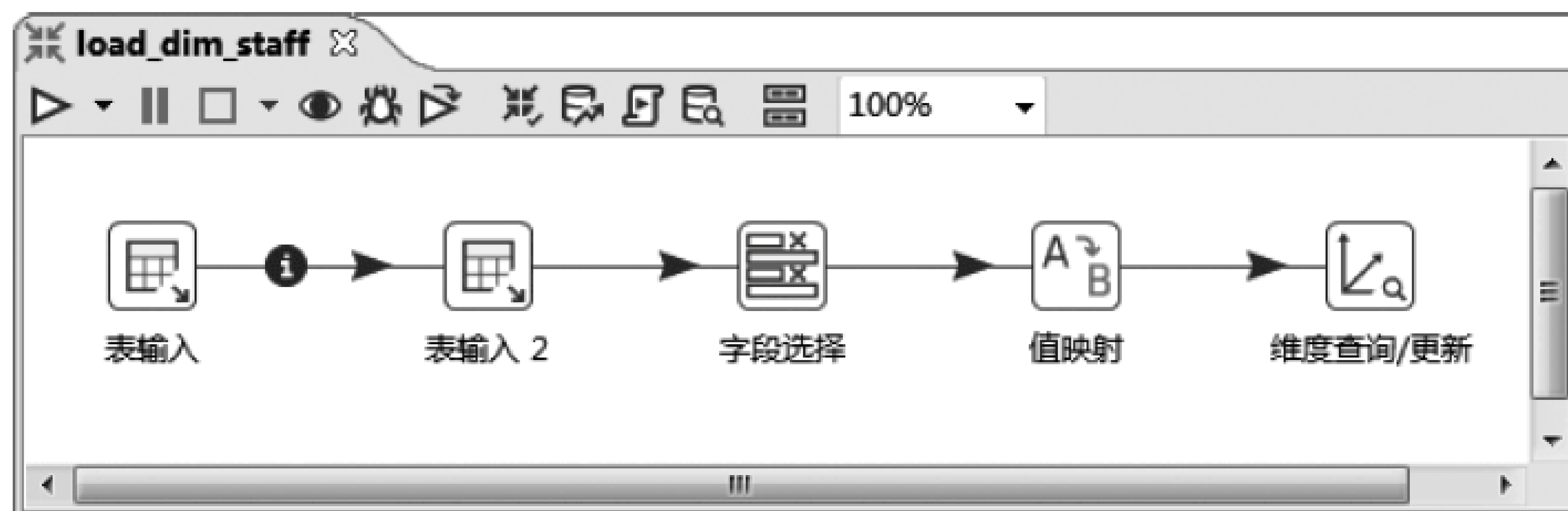


图 8-44 创建转换 load_dim_staff

2. 配置“表输入”控件

双击图 8-44 中的“表输入”控件，进入“表输入”界面，如图 8-45 所示。



图 8-45　“表输入”界面

在图 8-45 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-46 所示。



图 8-46　MySQL 数据库连接的配置

在图 8-45 的 SQL 框中编写 SQL 语句，用于获取字段 staff_last_update 中的最大值，将该值替换为 1970-01-01 00:00:00 并赋值给临时字段 max_dim_staff_last_update；单击“预览”按钮，查看临时字段 max_dim_staff_last_update 是否将默认值设置为 1970-01-01 00:00:00，具体如图 8-47 和图 8-48 所示。



图 8-47　编写 SQL 语句

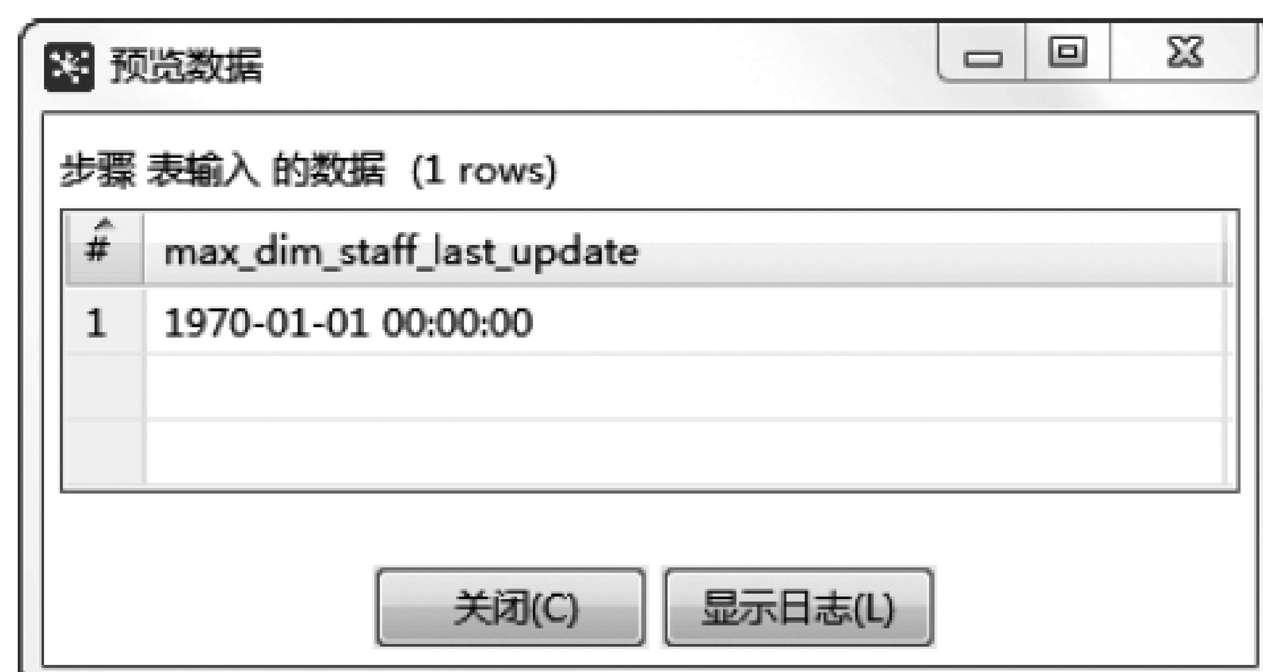


图 8-48　预览数据

从图 8-48 中可以看出，临时字段 max_dim_staff_last_update 的默认值设置为 1970-01-01 00:00:00，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“表输入 2”控件

双击图 8-44 中的“表输入 2”控件，进入“表输入”界面，如图 8-49 所示。



图 8-49　“表输入”界面

在图 8-49 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-50 所示。

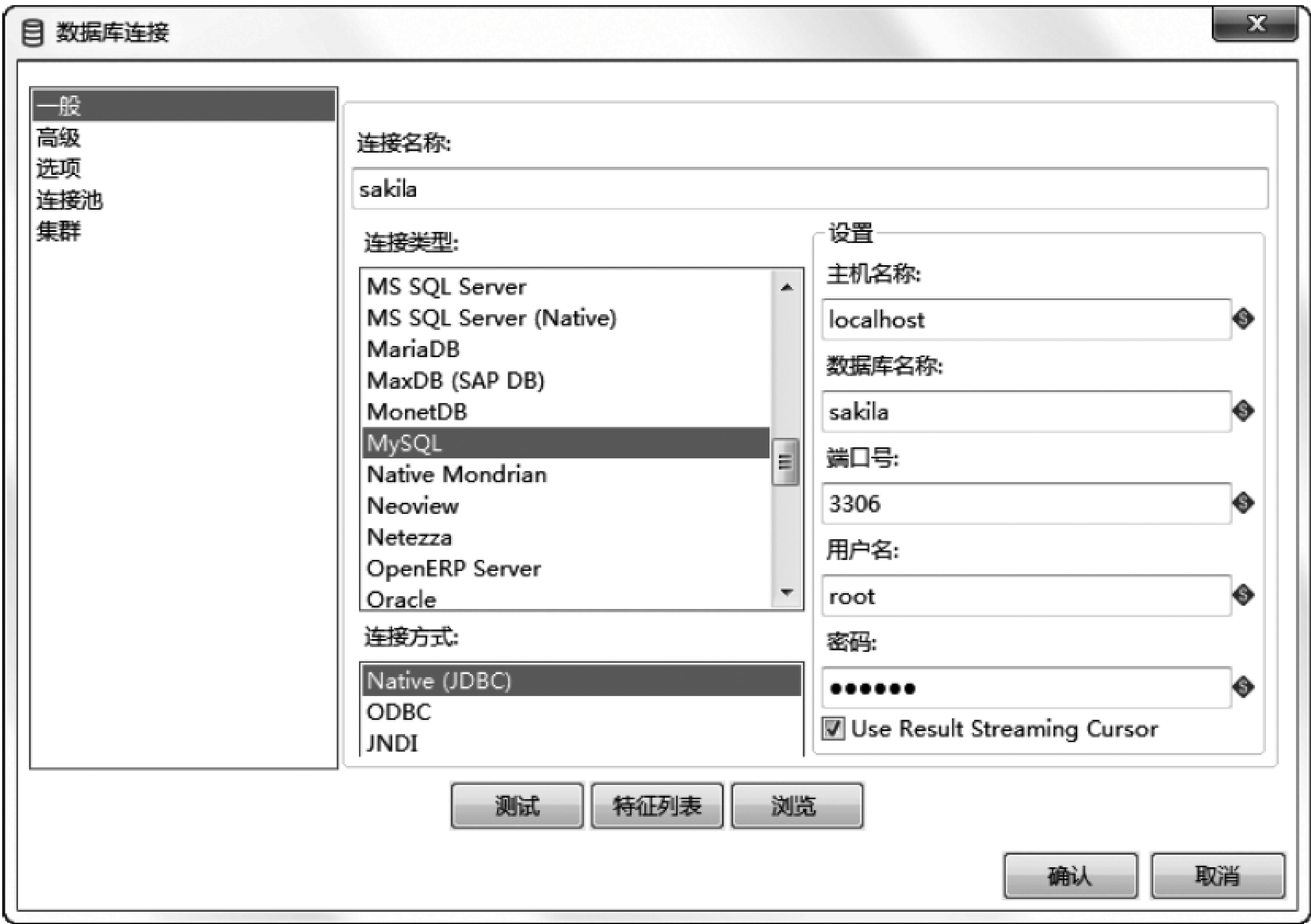


图 8-50　MySQL 数据库连接的配置

在图 8-49 的 SQL 框中编写 SQL 语句，用于获取数据库 sakila 中数据表 staff 中的最新数据，具体如图 8-51 所示。



图 8-51　编写 SQL 语句

在图 8-51 中单击“确定”按钮，完成“表输入 2”控件的配置。

4. 配置"字段选择"控件

双击图 8-44 中的"字段选择"控件,进入"选择/改名值"界面,在"元数据"选项卡的"需要改变元数据的字段"处添加字段 active,由于维度表 dim_staff 中字段 staff_active 的数据类型为 char 类型,因此需要将数据表 staff 中字段 active 的数据类型改为 String,具体如图 8-52 所示。



图 8-52 添加字段

在图 8-52 中单击"确定"按钮,完成"字段选择"控件的配置。

5. 配置"值映射"控件

双击图 8-44 中的"值映射"控件,进入"值映射"界面,在"使用的字段名"后的下拉列表中选择字段 active;在"字段值"框中添加源值和目标值,这里将 Y 替换成 Yes,将 N 替换成 No,具体如图 8-53 所示。



图 8-53 配置"值映射"控件

在图 8-53 中单击"确定"按钮,完成"值映射"控件的配置。

6. 配置"维度查询/更新"控件

双击图 8-44 中的"维度查询/更新"控件,进入"维度查询/更新"界面,具体如图 8-54 所示。

在图 8-54 中单击"新建"按钮,配置数据库连接,配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 8-55 所示。



图 8-54 “维度查询/更新”界面



图 8-55 MySQL 数据库连接的配置

在图 8-54 中单击目标表右侧的“浏览”按钮，选择输出的目标表，即维度表 dim_staff；在“关键字”选项卡处添加关键字字段 staff_id，用于指定维度表字段和“值映射”控件流中字段的比较条件，若维度表中的数据有更新，则通过字段 staff_id 进行更新操作，如图 8-56 所示；在“字段”选项卡处添加查询/更新字段，用于更新目标维度表中的字段数据，如图 8-57 所示；在“代理关键字段”后的下拉列表中选择 staff_key，并指定“创建代理键”为使用自增字段；在“Version 字段”后的下拉列表中选择 staff_version_number；在“Stream 日期字段”后的下拉列表中选择 last_update；在“开始日期字段”后的下拉列表中选择 staff_valid_from；在“截止日期字段”后的下拉列表中选择 staff_valid_through，具体如图 8-58 所示。



图 8-56 指定输出的目标表和添加关键字字段



图 8-57 添加查询/更新字段

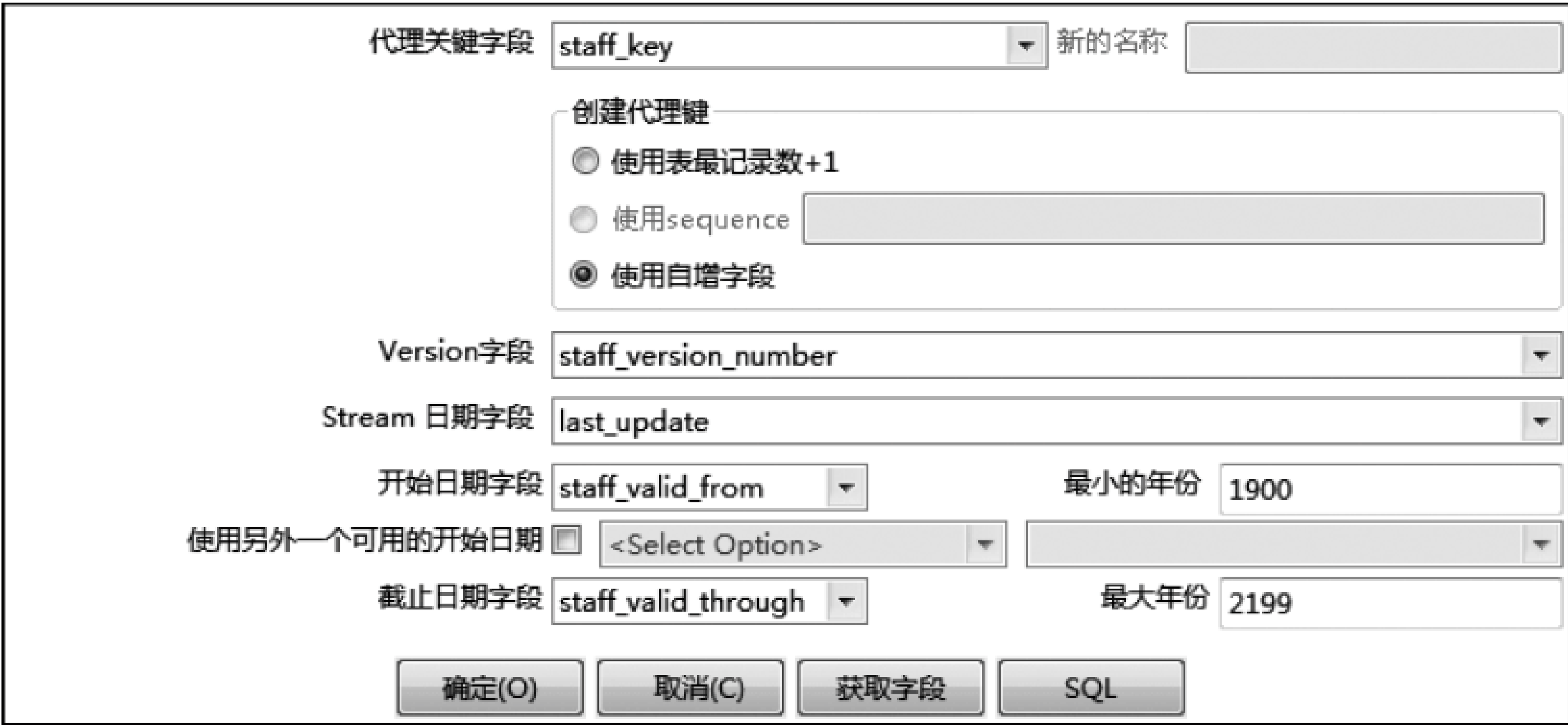
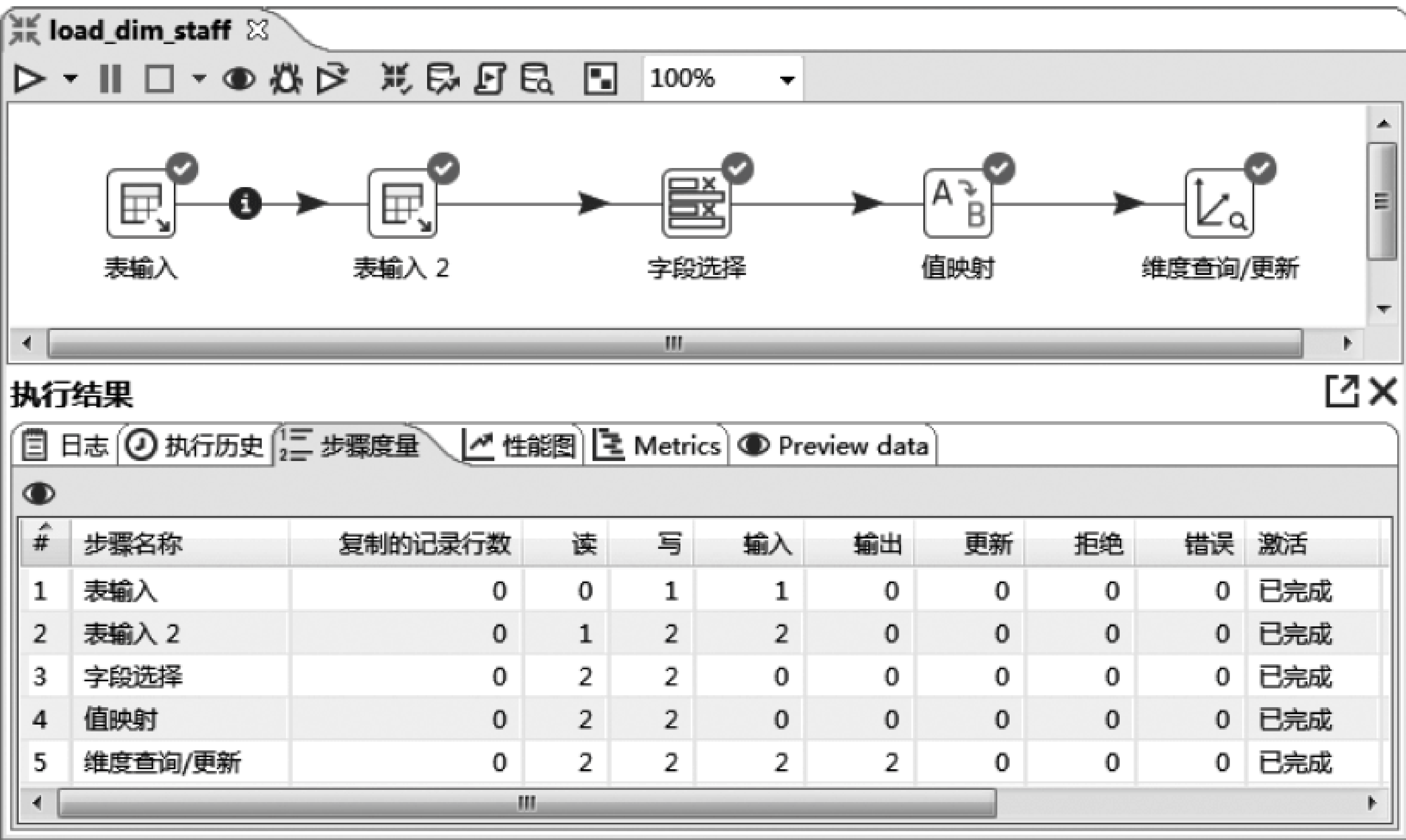


图 8-58 指定代理关键字段、Version 字段、Stream 日期字段、开始日期字段和截止日期字段

在图 8-58 中单击“确定”按钮，完成“维度查询/更新”控件的配置。

7. 运行转换 load_dim_staff

单击转换工作区顶部的▷按钮，运行创建的转换 load_dim_staff，实现加载员工数据至员工维度表 dim_staff 中，具体如图 8-59 所示。



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	表输入	0	0	1	1	0	0	0	0	已完成
2	表输入 2	0	1	2	2	0	0	0	0	已完成
3	字段选择	0	2	2	0	0	0	0	0	已完成
4	值映射	0	2	2	0	0	0	0	0	已完成
5	维度查询/更新	0	2	2	2	2	0	0	0	已完成

图 8-59 运行转换 load_dim_staff

从图 8-59 中执行结果的“步骤度量”可以看出，“表输入”控件输入 1 条数据并写入该控件；“表输入 2”控件从“表输入”控件中读取 1 条数据作为查询条件，将查询的 2 条数据作为输入并写入该控件；“字段选择”控件从“表输入 2”控件中读取 2 条数据并写入该控件；“值映射”控件从“字段选择”控件中读取 2 条数据并写入该控件；“维度查询/更新”控件输入 2 条数据并从“值映射”控件中读取 2 条数据，写入该控件 2 条数据，最终进行输出。

8. 查看维度表 dim_staff 中的数据

通过 SQLyog 工具，查看维度表 dim_staff 是否已成功插入员工数据，查看结果如图 8-60

所示。

	▲ sta...	staff_last_u...	staff_f...	staff_...	sta...	staff_...	staff_v...	staff_...	staff_va...	staff_active
□	8	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	1	(NULL)	(NULL)	(NULL)
□	9	1970-01-01 00:00	Mike	Hillyer	1	1	1	1900-01-01	2200-01-01	Yes
□	10	1970-01-01 00:00	Jon	Stephens	2	2	1	1900-01-01	2200-01-01	Yes
*	(Auto)	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

图 8-60　维度表 dim_staff

从图 8-60 中可以看出，维度表 dim_staff 中已插入数据，说明我们成功实现了加载员工数据至员工维度表 dim_staff 中。

8.3.5　加载用户数据至用户维度表

下面通过 Kettle 工具加载用户数据至用户维度表 dim_customer，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_customer，并添加“表输入”控件、“映射(子转换)”控件、“字段选择”控件、“值映射”控件、“维度查询/更新”控件以及 Hop 跳连接线，具体效果如图 8-61 所示。

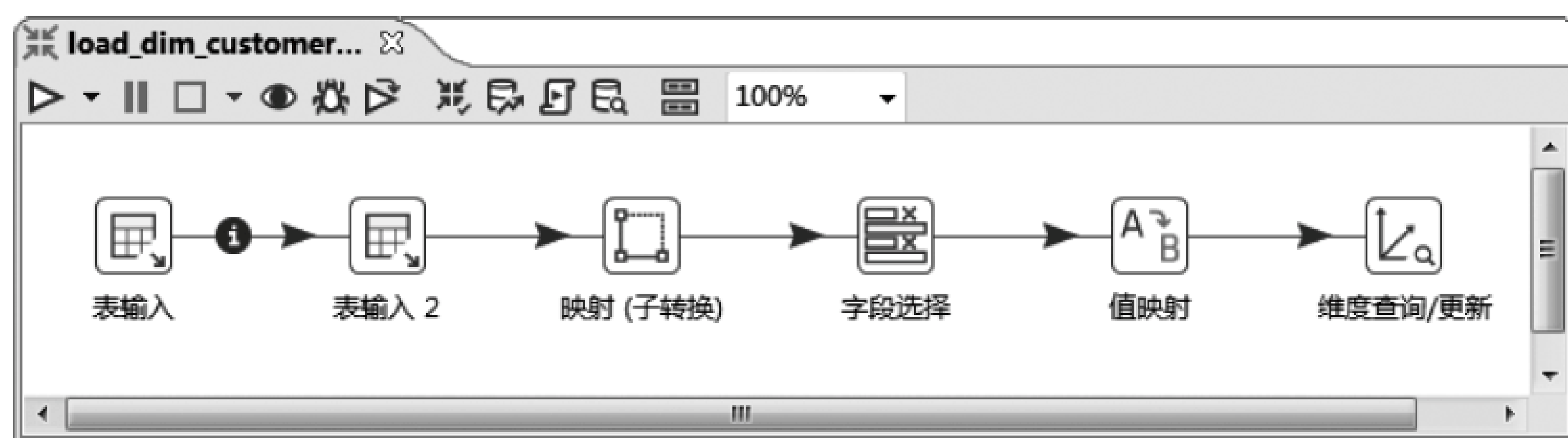


图 8-61　创建转换 load_dim_customer

2. 配置“表输入”控件

双击图 8-61 中的“表输入”控件，进入“表输入”界面，如图 8-62 所示。



图 8-62　“表输入”界面

在图 8-62 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-63 所示。



图 8-63　MySQL 数据库连接的配置

在图 8-62 的 SQL 框中编写用于获取字段 customer_last_update 中的最大值，将该值替换为 1970-01-01 00:00:00 并赋值给临时字段 max_dim_customer_last_update；单击“预览”按钮，查看临时字段 max_dim_customer_last_update 是否将默认值设置为 1970-01-01 00:00:00，具体如图 8-64 和图 8-65 所示。



图 8-64　编写 SQL 语句

从图 8-65 中可以看出，临时字段 max_dim_customer_last_update 的默认值设置为 1970-01-01 00:00:00，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。



图 8-65 预览数据

3. 配置“表输入 2”控件

双击图 8-61 中的“表输入 2”控件，进入“表输入”界面，如图 8-66 所示。



图 8-66 “表输入”界面

在图 8-66 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-67 所示。

在图 8-66 的 SQL 框中编写 SQL 语句，用于获取数据库 sakila 中数据表 customer 中的最新数据，具体如图 8-68 所示。

在图 8-68 中单击“确定”按钮，完成“表输入 2”控件的配置。这里需要注意的是，获取数据表 customer 的最新数据中包含字段 address_id，因此需要创建一个子转换，用于实现获取用户的地址，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 fetch_address(该转换为转换 load_dim_customer 的子转换)，并添加“映射输入规范”控件、“数据库查询”控件、“过滤记录”控件、“JavaScript 代码”控件、“字段选择”控件、“映射输出规范”控件以及 Hop 跳连接线，具体效果如图 8-69 所示。

2. 配置“映射输入规范”控件

双击图 8-69 中的“映射输入规范”控件，进入 Mapping input specification 界面，如图 8-70

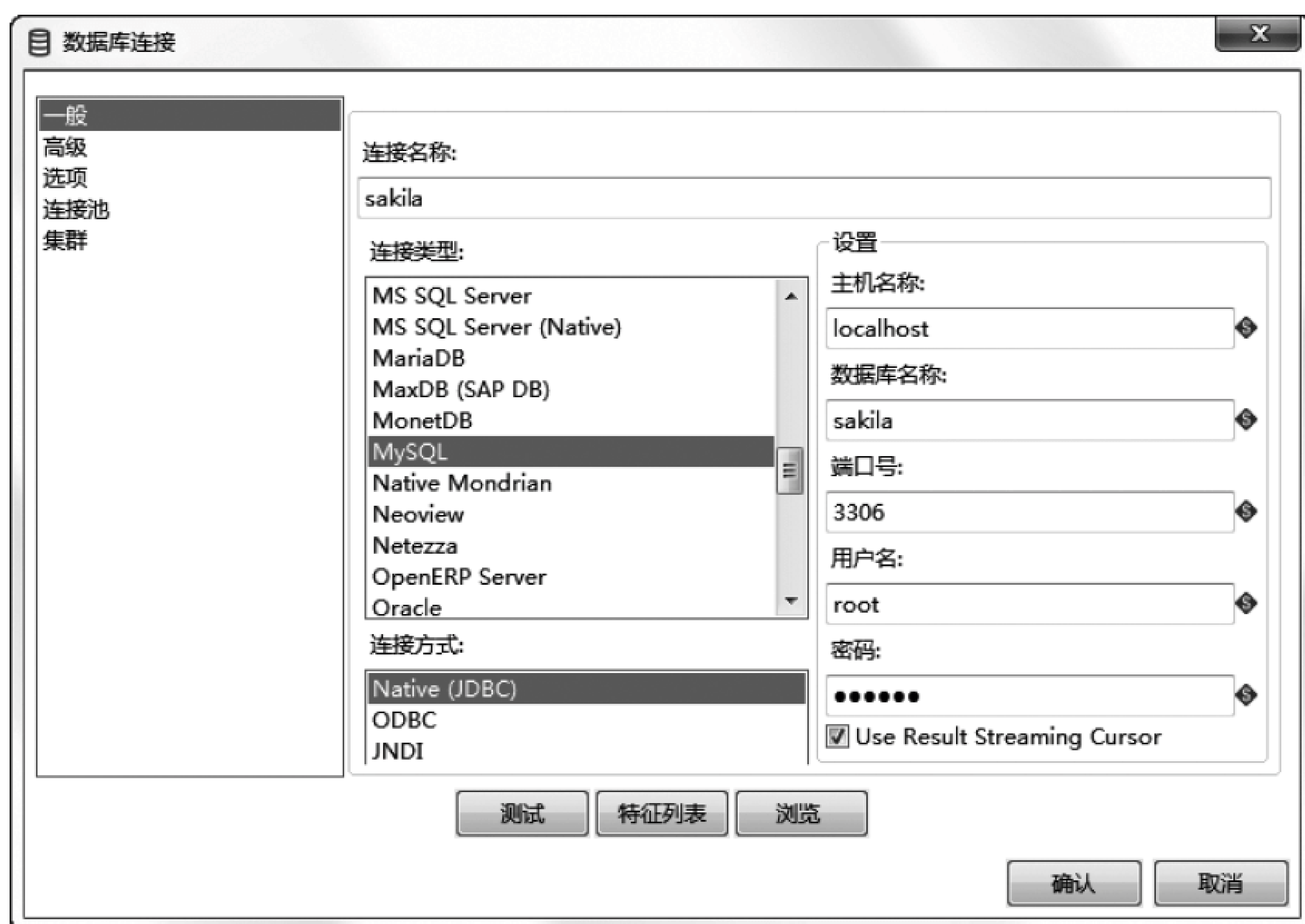


图 8-67　MySQL 数据库连接的配置



图 8-68　编写 SQL 语句

所示。

在图 8-70 中添加映射的字段 address_id，该字段为传递的参数（由于转换 load_dim_customer 中“表输入 2”控件流获取的字段 address_id 用于查询用户的地址信息，而后续数据仓库的维度表数据也需要用户的地址信息，因此这里将字段 address_id 作为传递的参数），添加完毕后单击“确定”按钮，完成“映射输入规范”控件的配置，具体如图 8-71 所示。

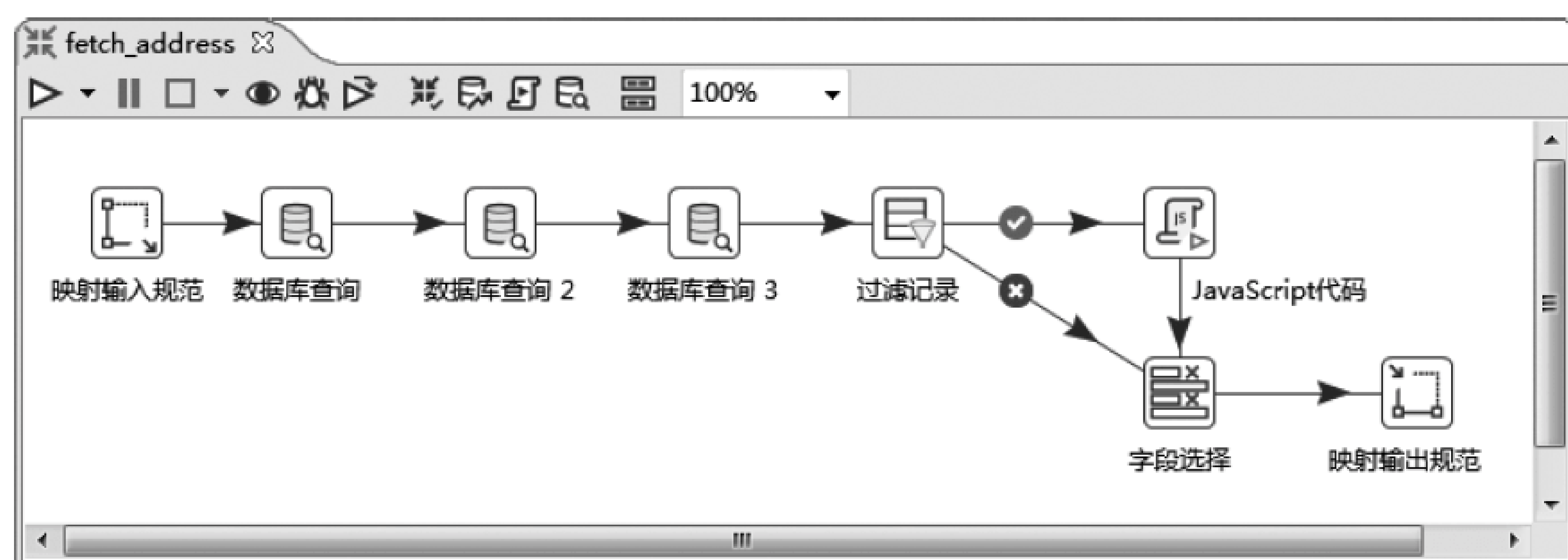


图 8-69 创建转换 fetch_address



图 8-70 Mapping input specification 界面



图 8-71 配置“映射输入规范”控件

3. 配置“数据库查询”控件

双击图 8-69 中的“数据库查询”控件，进入“数据库查询”界面，如图 8-72 所示。

在图 8-72 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-73 所示。

单击图 8-72 中表名右侧的“浏览”按钮，添加数据表 address；在“查询所需的关键字”框中添加查询所需的关键字字段 address_id，由于该字段是唯一的，因此可作为数据表 address 中数据和“映射输入规范”控件流中数据的比较条件；在“查询表返回的值”框中添加查询表返回的值，如图 8-74 所示。

在图 8-74 中单击“确定”按钮，完成“数据库查询”控件的配置。

4. 配置“数据库查询 2”控件

双击图 8-69 中的“数据库查询 2”控件，进入“数据库查询”界面，如图 8-75 所示。



图 8-72　“数据库查询”界面

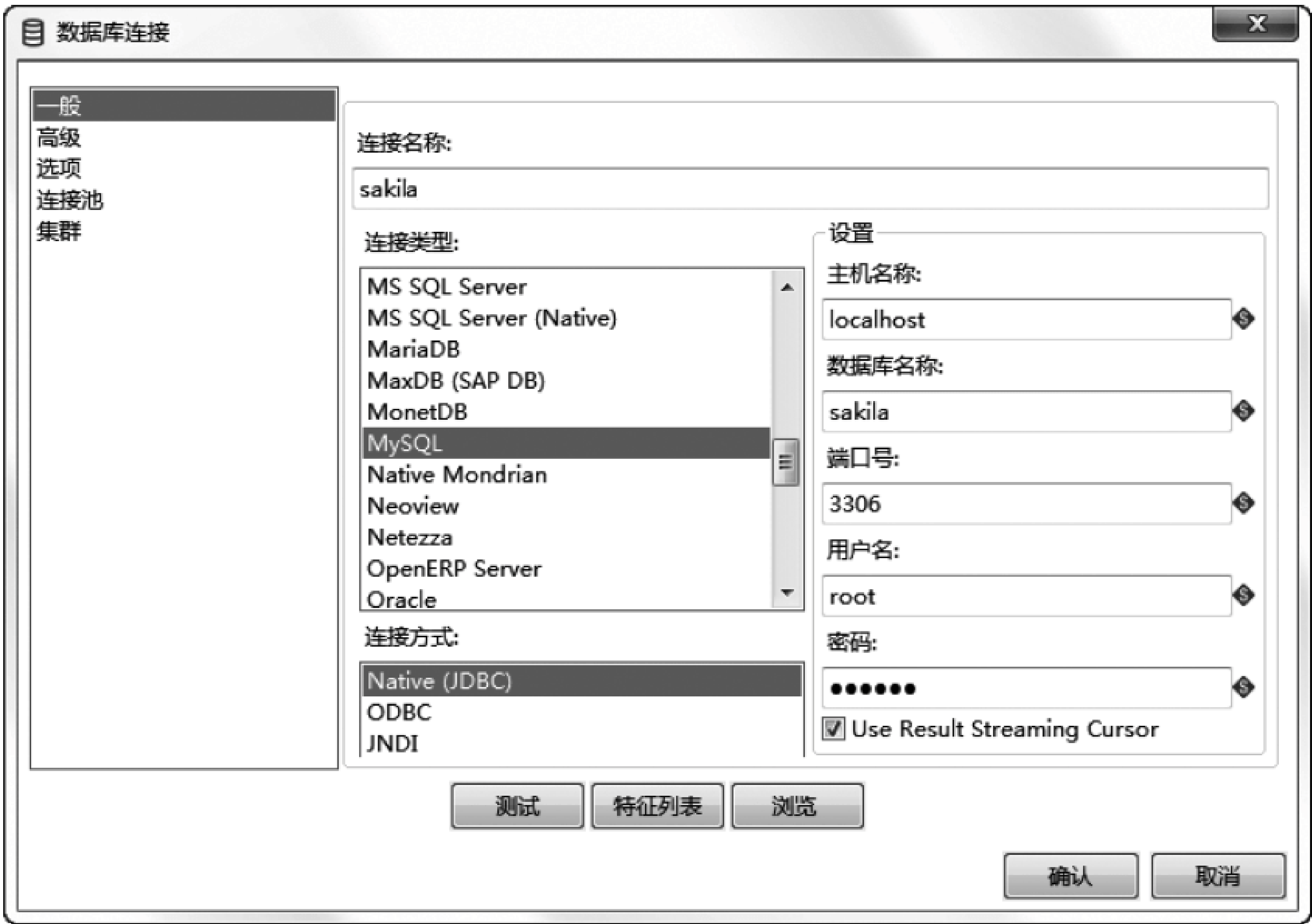


图 8-73　MySQL 数据库连接的配置

在图 8-75 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-76 所示。

单击图 8-75 中表名后的“浏览”按钮，添加数据表 city；在“查询所需的关键字”框中添加查询所需的关键字字段 city_id，作为数据表 city 中数据与“数据库查询 2”控件流中数据比较的条件；在“查询表返回的值”框中添加查询表返回的值，如图 8-77 所示。

图 8-74 配置“数据库查询”控件

图 8-75 “数据库查询”界面

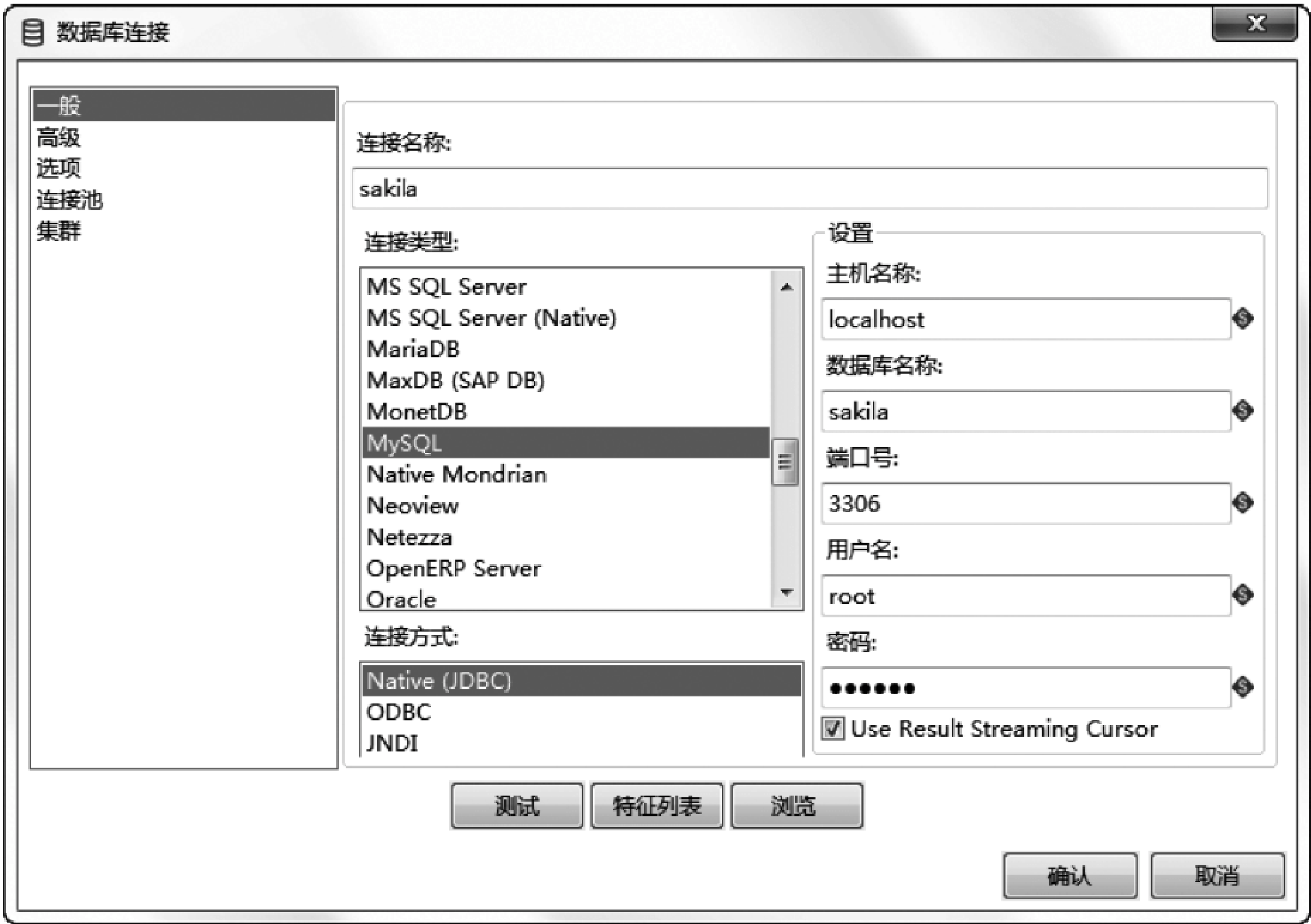


图 8-76　MySQL 数据库连接的配置



图 8-77　配置“数据库查询 2”控件

在图 8-77 中单击“确定”按钮，完成“数据库查询 2”控件的配置。

5. 配置“数据库查询 3”控件

双击图 8-69 中的“数据库查询 3”控件，进入“数据库查询”界面，如图 8-78 所示。

图 8-78 “数据库查询”界面

在图 8-78 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-79 所示。

图 8-79 MySQL 数据库连接的配置

单击图 8-78 中表名后的“浏览”按钮，添加数据表 country；在“查询所需的关键字”框中添加查询所需的关键字字段 country_id，用作指定字段流与表字段的数据进行比较的条件；在“查询表返回的值”框中添加查询表返回的值，如图 8-80 所示。



图 8-80　配置“数据库查询 3”控件

在图 8-80 中单击“确定”按钮，完成“数据库查询 3”控件的配置。

6. 配置“过滤记录”控件

双击图 8-69 中的“过滤记录”控件，进入“过滤记录”界面，如图 8-81 所示。



图 8-81　“过滤记录”界面

在图 8-81 中的“条件”处设置过滤的条件，对有第二个地址的用户进行过滤操作；单击左边的＜field＞框，弹出“字段”对话框，选择要过滤的字段 address2，如图 8-82 所示。

在图 8-82 中单击“确定”按钮，完成过滤字段 address2 的选择。

单击图 8-81 中的“＝”框，弹出“函数：”对话框，选择过滤条件（这里选择的是 IS NOT NULL），即过滤指定字段中不为空的数据，如图 8-83 所示。



图 8-82 “字段”对话框



图 8-83 “函数:”对话框

在图 8-83 中单击“确定”按钮,完成过滤条件的选择。字段 address2 的过滤设置如图 8-84 所示。

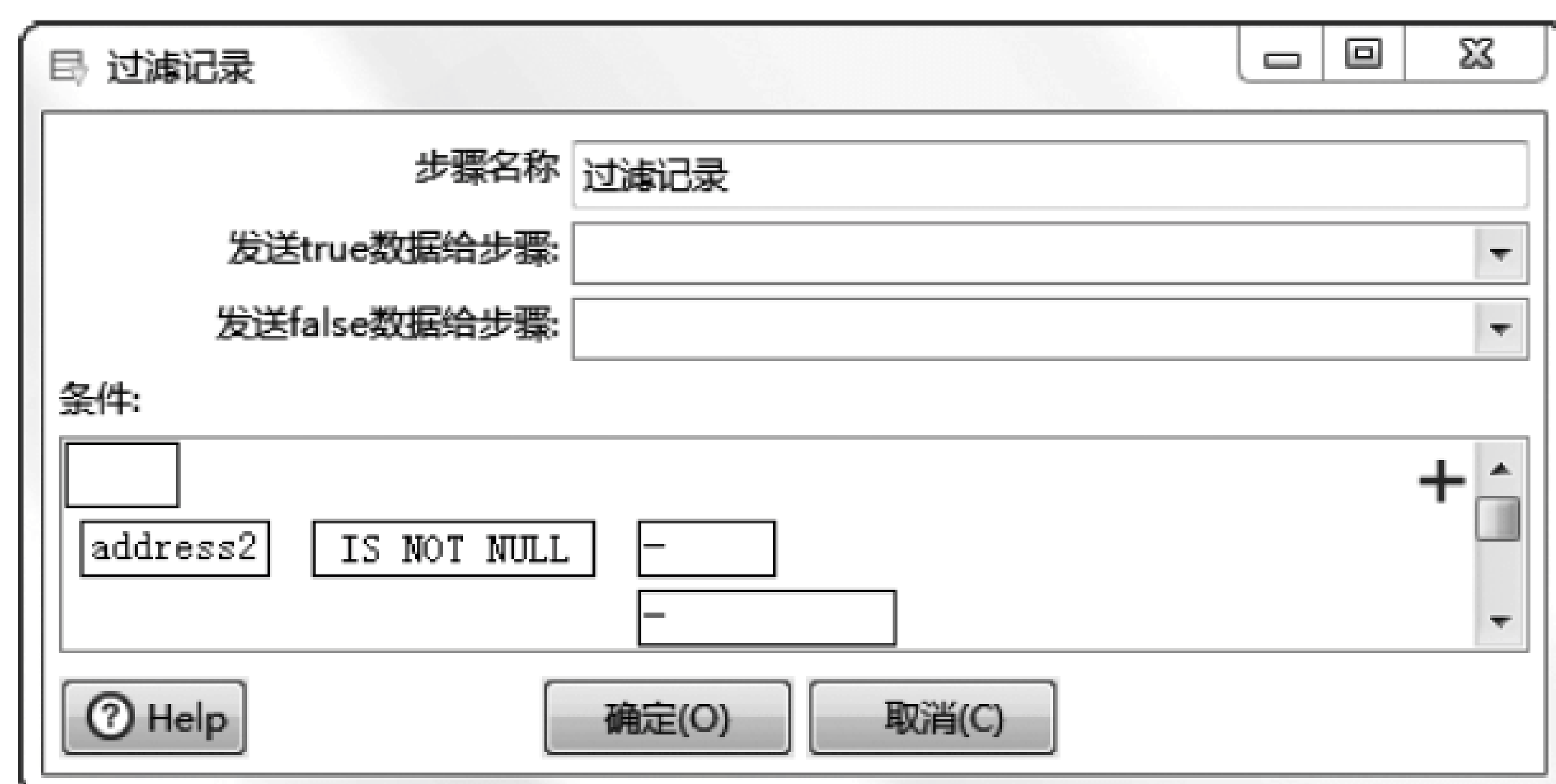


图 8-84 字段 address2 的过滤设置

在图 8-84 中“发送 true 数据给步骤:”后的下拉列表中选择“JavaScript 代码”,将有第二个地址的用户放在 JavaScript 控件中,用于后续的操作;在“发送 false 数据给步骤:”后的下拉列表中选择“字段选择”,将没有第二个地址的用户进行字段选择处理,具体如图 8-85 所示。



图 8-85 配置“过滤记录”控件

在图 8-85 中单击“确定”按钮，完成“过滤记录”控件的配置。

7. 配置“JavaScript 代码”控件

双击图 8-69 中的“JavaScript 代码”控件，进入“JavaScript 代码”界面，勾选“兼容模式?”复选框，使得“JavaScript 代码”控件的兼容性更强；在 JavaScript 代码框中编写代码，具体如图 8-86 所示。



图 8-86 配置“JavaScript 代码”控件

在图 8-86 中单击“确定”按钮，完成“JavaScript 代码”控件的配置。

8. 配置“字段选择”控件

双击图 8-69 中的“字段选择”控件，进入“选择/改名值”界面，在“移除”选项卡的“移除的字段:”处添加要移除的字段，具体如图 8-87 所示。



图 8-87 添加字段

在图 8-87 中单击“确定”按钮，完成“字段选择”控件的配置。

9. 配置"映射"控件

双击图 8-61 中的"映射(子转换)"控件,进入"映射(执行子转换任务)"界面,单击"转换"处的 Browser 按钮,选择添加转换 fetch_address,用于获取用户的地址信息,具体如图 8-88 所示。



图 8-88　添加转换 fetch_address

在图 8-88 中单击"确定"按钮,完成"映射"控件的配置。

10. 配置"字段选择"控件

双击图 8-61 中的"字段选择"控件,进入"选择/改名值"界面,在"元数据"选项卡的"需要改变元数据的字段"处添加字段 active,由于数据表 customer 中字段 active 的类型为 tinyint,因此需要将字段 active 的类型改为 String,与维度表 dim_customer 中字段 customer_active 的类型对应,具体如图 8-89 所示。



图 8-89　添加字段 active

在图 8-89 中单击"确定"按钮,完成"字段选择"控件的配置。

11. 配置“值映射”控件

双击图 8-61 中的“值映射”控件，进入“值映射”界面，在“使用的字段名：”后的下拉列表中选择字段 active；在“字段值”框中添加源值和目标值，由于数据表 customer 中字段 active 的值为 1 和 0，对应的是 Y 和 N，这里将 Y 替换成 Yes，将 N 替换成 No，具体如图 8-90 所示。



图 8-90　配置“值映射”控件

在图 8-90 中单击“确定”按钮，完成“值映射”控件的配置。

12. 配置“维度查询/更新”控件

双击图 8-61 中的“维度查询/更新”控件，进入“维度查询/更新”界面，具体如图 8-91 所示。

在图 8-91 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-92 所示。

在图 8-91 中单击目标表右侧的“浏览”按钮，选择输出的目标表，即维度表 dim_customer；在“关键字”选项卡处添加关键字字段 customer_id，用于指定维度表字段和流字段的比较条件，如图 8-93 所示；在“字段”选项卡处添加查询/更新字段，如图 8-94 所示；在“代理关键字段”后的下拉列表中选择 customer_key 为代理关键字段，并指定“创建代理键”为使用自增字段；在“Version 字段”后的下拉列表中选择 customer_version_number；在“Stream 日期字段”后的下拉列表中选择 last_update；在“开始日期字段”后的下拉列表中选择 customer_valid_from；在“截止日期字段”后的下拉列表中选择 customer_valid_through，具体如图 8-95 所示。

在图 8-95 中单击“确定”按钮，完成“维度查询/更新”控件的配置。

13. 运行转换 load_dim_customer

单击转换工作区顶部的▷按钮，运行创建的转换 load_dim_customer，实现加载用户数据至用户维度表 dim_customer 中，具体如图 8-96 所示。

从图 8-96 中执行结果的“步骤度量”可以看出，“表输入”控件输入 1 条数据并写入该控件；“表输入 2”控件输入 599 条数据并从“表输入”控件中读取 1 条输入，写入该控件 599 条



图 8-91 “维度查询/更新”界面



图 8-92 MySQL 数据库连接的配置



图 8-93　指定输出的目标表和添加关键字字段

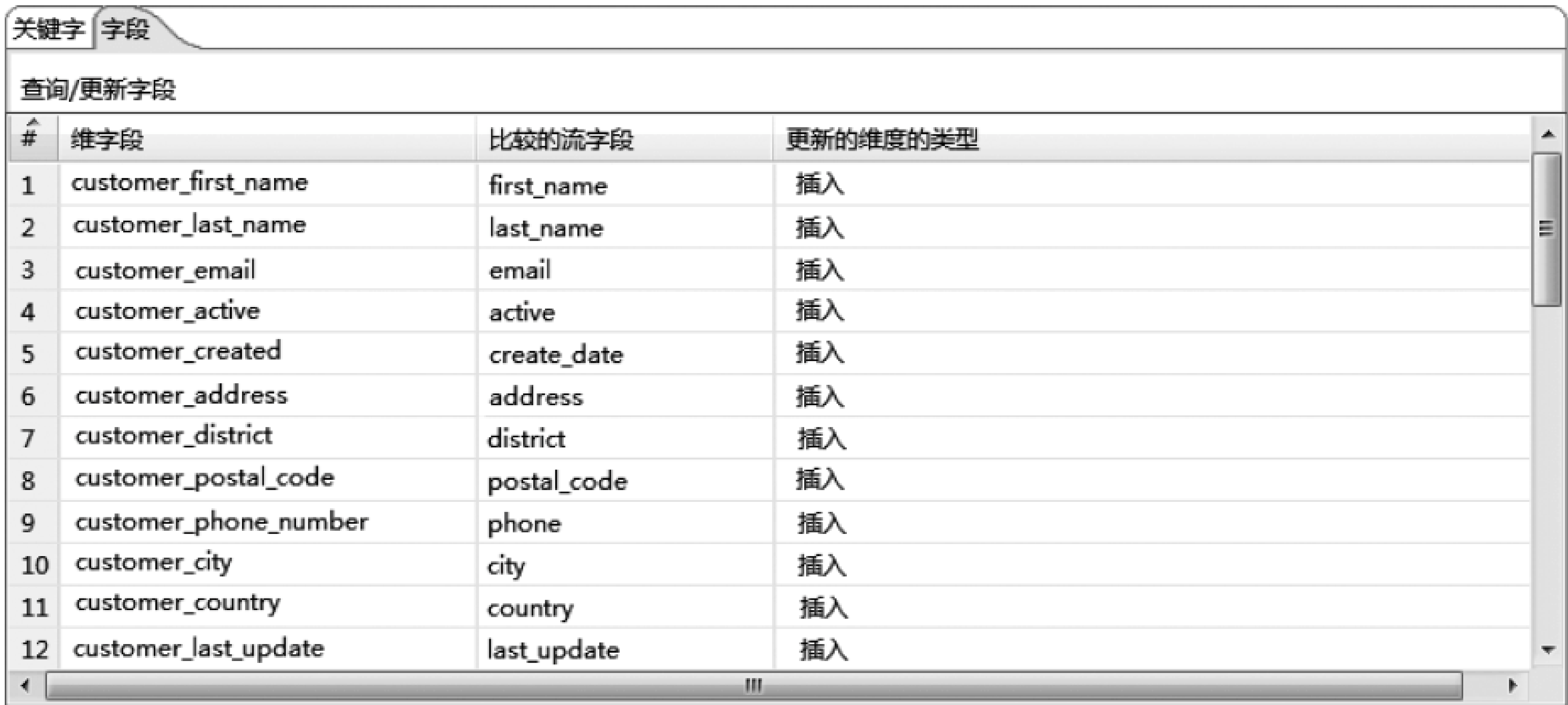


图 8-94　添加查询/更新字段

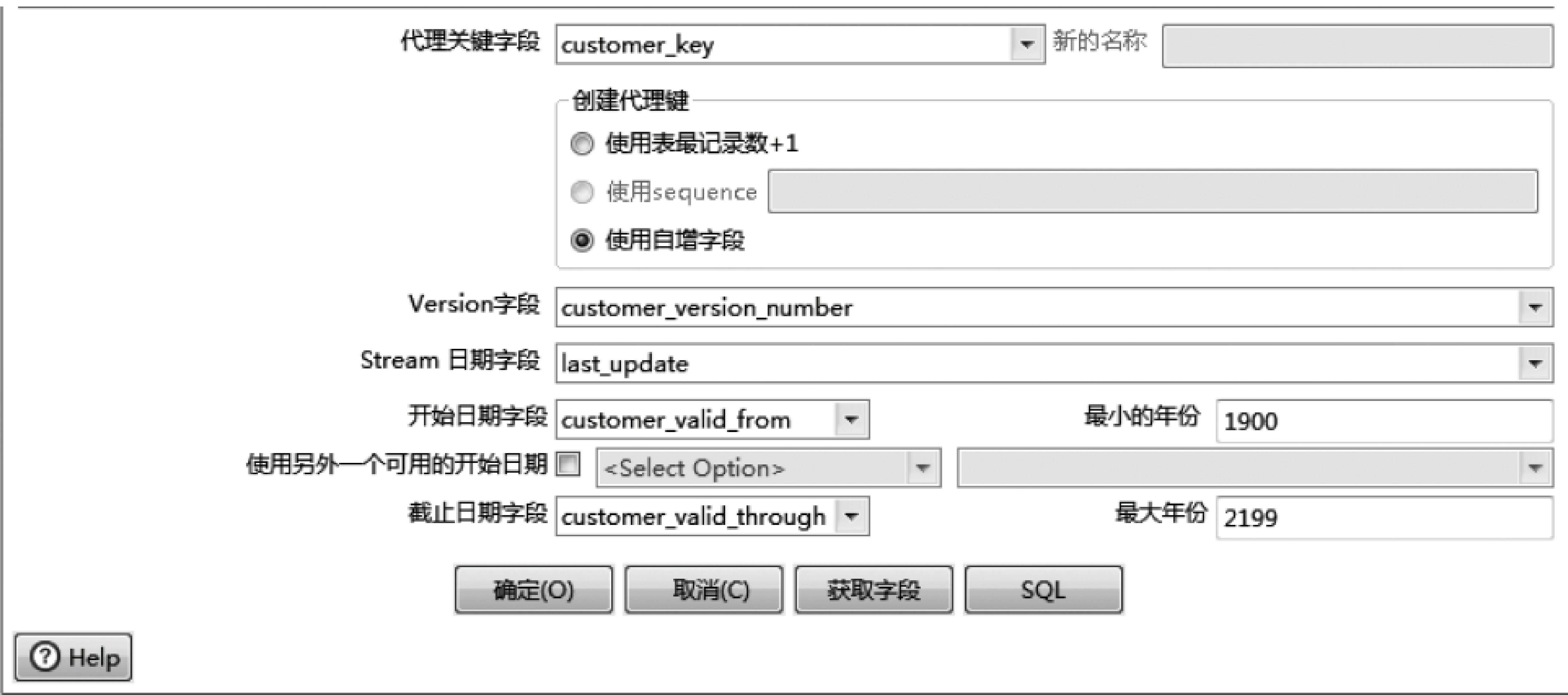
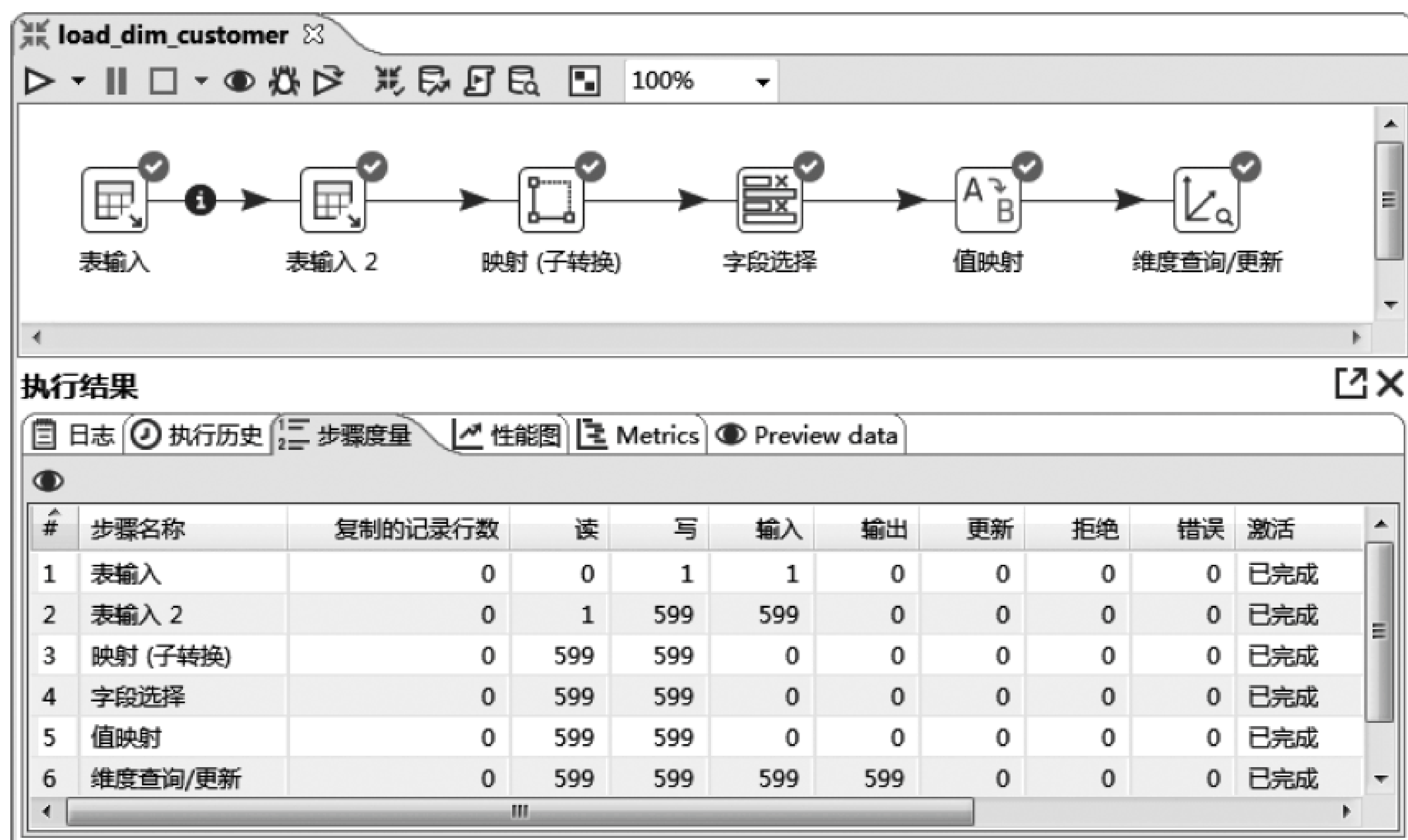


图 8-95　指定代理关键字段、Version 字段、Stream 日期字段、开始日期字段和截止日期字段



#	步骤名称	复制的记录行数	读	写	输入	输出	更新	拒绝	错误	激活
1	表输入	0	0	1	1	0	0	0	0	已完成
2	表输入 2	0	1	599	599	0	0	0	0	已完成
3	映射 (子转换)	0	599	599	0	0	0	0	0	已完成
4	字段选择	0	599	599	0	0	0	0	0	已完成
5	值映射	0	599	599	0	0	0	0	0	已完成
6	维度查询/更新	0	599	599	599	599	0	0	0	已完成

图 8-96　运行转换 load_dim_customer

数据；“映射(子转换)”控件从“表输入 2”控件中读取 599 条数据并写入该控件；“字段选择”控件从“映射(子转换)”控件中读取 599 条数据并写入该控件；“值映射”控件从“字段选择”控件中读取 599 条数据并写入该控件；“维度查询/更新”控件输入 599 条数据，并从“值映射”控件中读取 599 条数据，写入该控件，最终进行输出。

14．查看维度表 dim_customer 中的数据

通过 SQLyog 工具，查看维度表 dim_customer 是否已成功插入用户数据，查看结果如图 8-97 所示(这里只展示部分数据)。

cust...	customer_last_up...	c...	custo...	cus...	customer_email	cu...	custome...	customer_address	customer_district
2398	2006-02-15 04:57:20	1	MARY	SMITH	MARY.SMITH@saki	Yes	2006-02-14	1913 Hanoi Way	Nagasaki
2399	2006-02-15 04:57:20	2	PATRICIA	JOHNSON	PATRICIA.JOHNSC	Yes	2006-02-14	1121 Loja Avenue	California
2400	2006-02-15 04:57:20	3	LINDA	WILLIAM	LINDA.WILLIAMS@	Yes	2006-02-14	692 Joliet Street	Attika
2401	2006-02-15 04:57:20	4	BARBARA	JONES	BARBARA.JONES@s	Yes	2006-02-14	1566 Inegl Manor	Mandalay
2402	2006-02-15 04:57:20	5	ELIZABET	BROWN	ELIZABETH.BROWN	Yes	2006-02-14	53 Idfu Parkway	Nantou
2403	2006-02-15 04:57:20	6	JENNIFER	DAVIS	JENNIFER.DAVIS@	Yes	2006-02-14	1795 Santiago de	Texas
2404	2006-02-15 04:57:20	7	MARIA	MILLER	MARIA.MILLER@sa	Yes	2006-02-14	900 Santiago de C	Central Serbia
2405	2006-02-15 04:57:20	8	SUSAN	WILSON	SUSAN.WILSON@sa	Yes	2006-02-14	478 Joliet Way	Hamilton
2406	2006-02-15 04:57:20	9	MARGARET	MOORE	MARGARET.MOORE@	Yes	2006-02-14	613 Korolev Drive	Masqat

图 8-97　维度表 dim_customer

从图 8-97 中可以看出，维度表 dim_customer 中已插入数据，说明我们成功实现了加载用户数据至用户维度表 dim_customer。

8.3.6　加载商店数据至商店维度表

下面通过 Kettle 工具加载商店数据至商店维度表 dim_store，具体实现步骤如下。

1．打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_store，并添加“表输入”控件、“映射(子转换)”控

件、“数据库查询”控件、“维度查询/更新”控件以及 Hop 跳连接线，具体效果如图 8-98 所示。

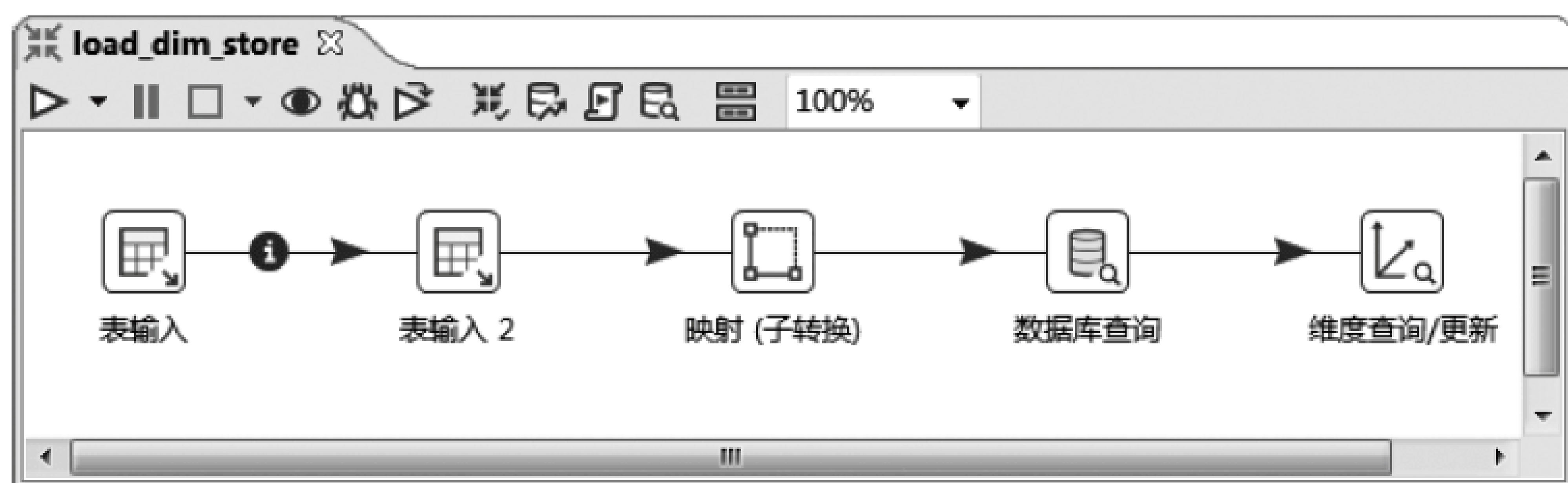


图 8-98　创建转换 load_dim_store

2. 配置“表输入”控件

双击图 8-98 中的“表输入”控件，进入“表输入”界面，如图 8-99 所示。



图 8-99　“表输入”界面

在图 8-99 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-100 所示。

在图 8-99 的 SQL 框中编写 SQL 语句，用于获取字段 store_last_update 中的最大值，将该值替换为 1970-01-01 00:00:00 并赋值给临时字段 max_dim_store_last_update；单击“预览”按钮，查看临时字段 max_dim_store_last_update 是否将默认值设置为 1970-01-01 00:00:00，具体如图 8-101 和图 8-102 所示。

从图 8-102 中可以看出，临时字段 max_dim_store_last_update 的默认值设置为 1970-01-01 00:00:00，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“表输入 2”控件

双击图 8-98 中的“表输入 2”控件，进入“表输入”界面，如图 8-103 所示。

在图 8-103 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-104 所示。



图 8-100　MySQL 数据库连接的配置



图 8-101　编写 SQL 语句



图 8-102　预览数据



图 8-103　“表输入”界面

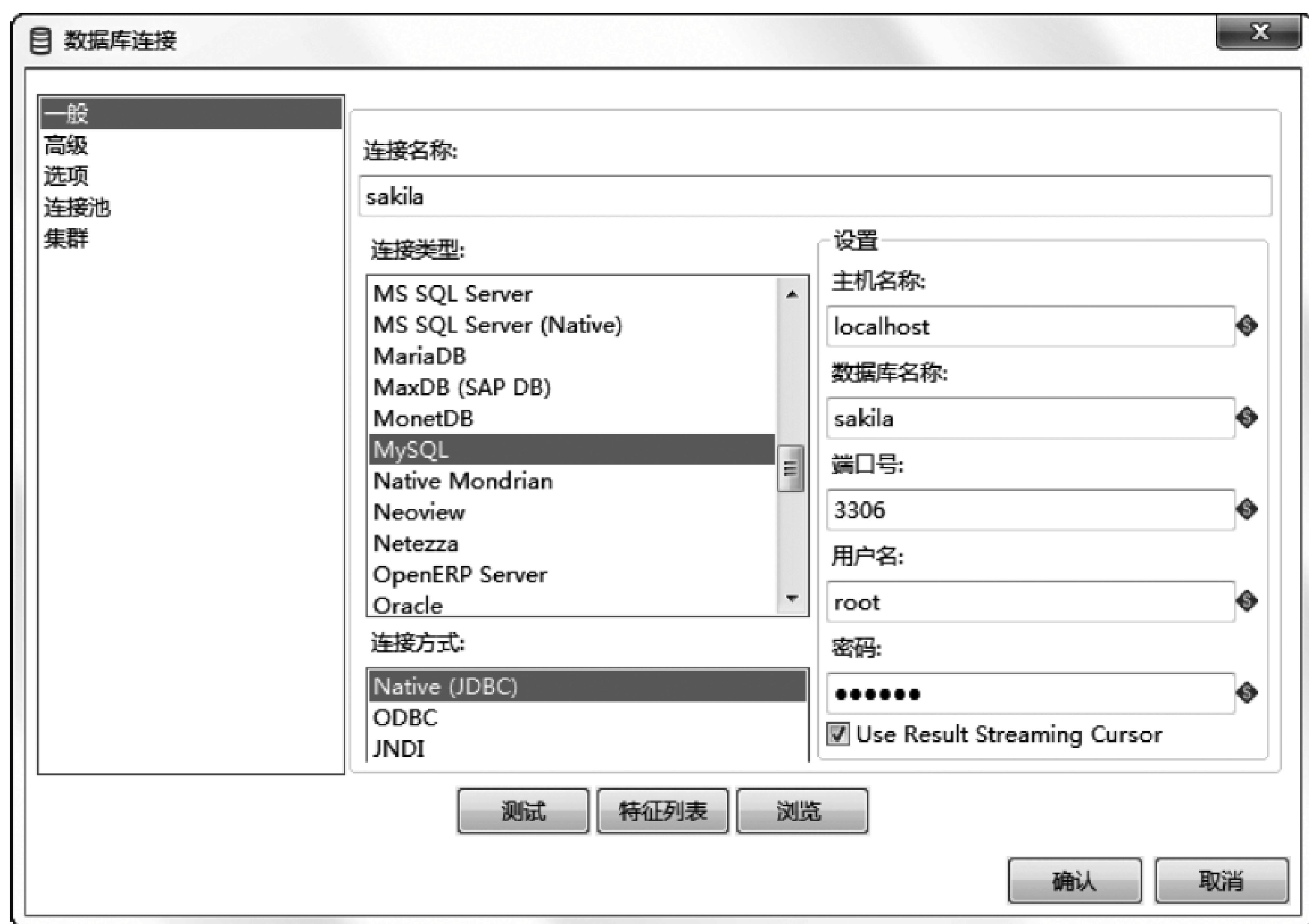


图 8-104　MySQL 数据库连接的配置

在图 8-103 的 SQL 框中编写 SQL 语句，用于获取 sakila 数据库中 store 数据表中的最新数据，具体如图 8-105 所示。

在图 8-105 中单击“确定”按钮，完成“表输入 2”控件的配置。

4. 配置“映射”控件

双击图 8-98 中的“映射(子转换)”控件，进入“映射(执行子转换任务)”界面，单击“转换”处的 Browser 按钮，选择添加转换 fetch_address，用于获取用户的地址信息，如图 8-106 所示。

在图 8-106 中单击“确定”按钮，完成“映射(子转换)”控件的配置。



图 8-105　编写 SQL 语句



图 8-106　添加转换 fetch_address

5. 配置"数据库查询"控件

双击图 8-98 中的"数据库查询"控件，进入"数据库查询"界面，如图 8-107 所示。

在图 8-107 中单击"新建"按钮，配置数据库连接，配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 8-108 所示。

单击图 8-107 中表名右侧的"浏览"按钮，添加数据表 staff，用于查询商店员工的信息；在"查询所需的关键字"框中添加查询所需的关键字 staff_id，用于指定表字段和流字段的比较条件；在"查询表返回的值"框中添加查询表返回的值，即员工姓氏和名字，如图 8-109 所示。

在图 8-109 中单击"确定"按钮，完成"数据库查询"控件的配置。



图 8-107　“数据库查询”界面

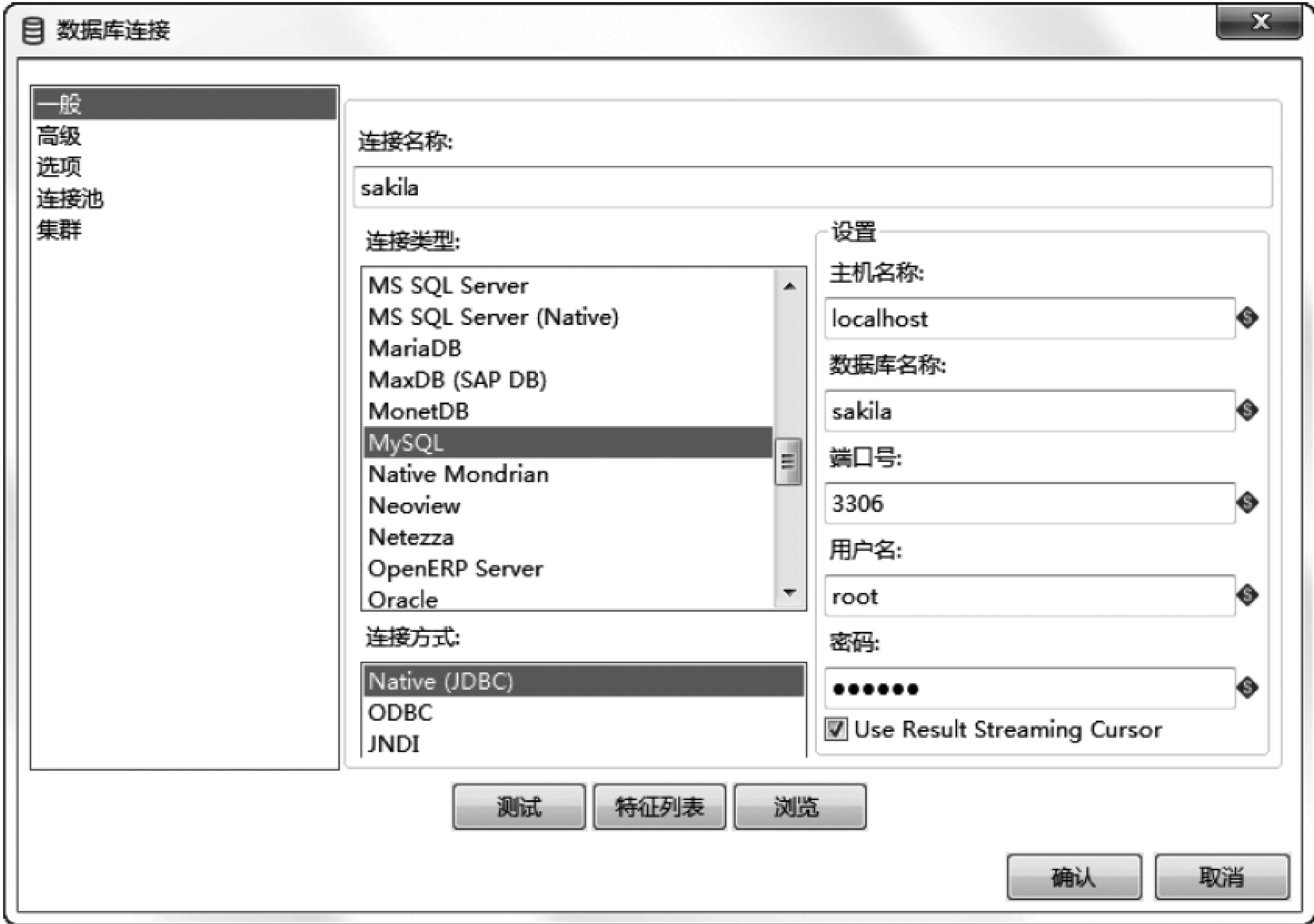


图 8-108　MySQL 数据库连接的配置

6. 配置“维度查询/更新”控件

双击图 8-98 中的“维度查询/更新”控件，进入“维度查询/更新”界面，具体如图 8-110 所示。

在图 8-110 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-111 所示。

图 8-109 配置“数据库查询”控件

图 8-110 “维度查询/更新”界面



图 8-111　MySQL 数据库连接的配置

在图 8-110 中单击“浏览”按钮，选择输出的目标表，即维度表 dim_store；在“关键字”选项卡中添加关键字字段 store_id，用于指定维度表字段和流字段的比较条件，如图 8-112 所示；在“字段”选项卡中添加查询/更新字段，用于指定维度表字段 store_id 和流字段 store_id 数据一致需要更新的字段，如图 8-113 所示；在“代理关键字段”后的下拉列表中选择 store_key，并指定“创建代理键”为使用自增字段；在“Version 字段”后的下拉列表中选择 store_version_number；在“Stream 日期字段”后的下拉列表中选择 last_update；在“开始日期字段”后的下拉列表中选择 store_valid_from；在“截止日期字段”后的下拉列表中选择 store_valid_through，具体如图 8-114 所示。

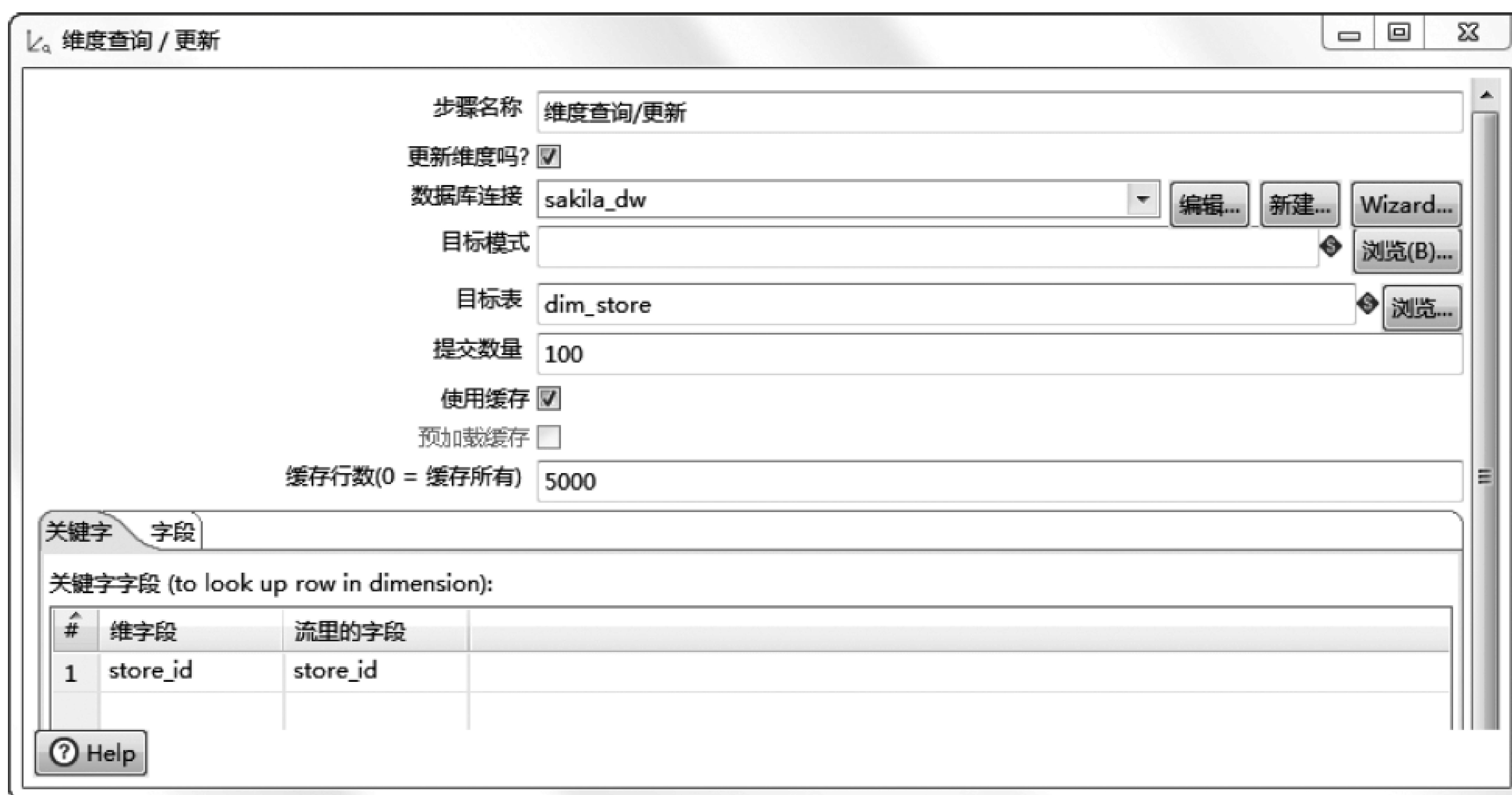


图 8-112　指定输出的目标表和添加关键字字段

关键字 字段

查询/更新字段

#	维字段	比较的流字段	更新的维度的类型
1	store_manager_staff_id	manager_staff_id	插入
2	store_last_update	last_update	插入
3	store_address	address	插入
4	store_district	district	插入
5	store_postal_code	postal_code	插入
6	store_phone_number	phone	插入
7	store_city	city	插入
8	store_country	country	插入
9	store_manager_first_name	first_name	插入
10	store_manager_last_name	last name	插入

图 8-113 添加查询/更新字段

代理关键字段 store_key 新的名称
创建代理键
使用表最记录数+1
使用sequence
使用自增字段
Version字段 store_version_number
Stream 日期字段 last_update
开始日期字段 store_valid_from 最小的年份 1900
使用另外一个可用的开始日期 <Select Option>
截止日期字段 store_valid_through 最大年份 2199
确定(O) 取消(C) 获取字段 SQL
Help

图 8-114 指定代理关键字段、Version 字段、Stream 日期字段、开始日期字段和截止日期字段

在图 8-114 中单击“确定”按钮，完成“维度查询/更新”控件的配置。

7. 运行转换 load_dim_store

单击转换工作区顶部的▷按钮，运行创建的转换 load_dim_store，实现加载商店数据至商店维度表 dim_store，具体如图 8-115 所示。

从图 8-115 中执行结果的“步骤度量”可以看出，“表输入”控件输入 1 条数据并写入该控件；“表输入 2”控件输入 2 条数据，并从“表输入”控件中读取 1 条数据，写入该控件 2 条数据；“映射(子转换)”控件从“表输入 2”控件中读取 2 条数据并写入该控件；“数据库查询”控件输入 2 条数据，并从“映射(子转换)”控件中读取 2 条数据，写入该控件 2 条数据；“维度查询/更新”控件输入 2 条数据，并从“数据库查询”控件中读取 2 条数据并写入该控件，最终进行输出。

8. 查看维度表 dim_store 中的数据

通过 SQLyog 工具，查看维度表 dim_store 是否已成功插入商店数据，查看结果如图 8-116 所示。

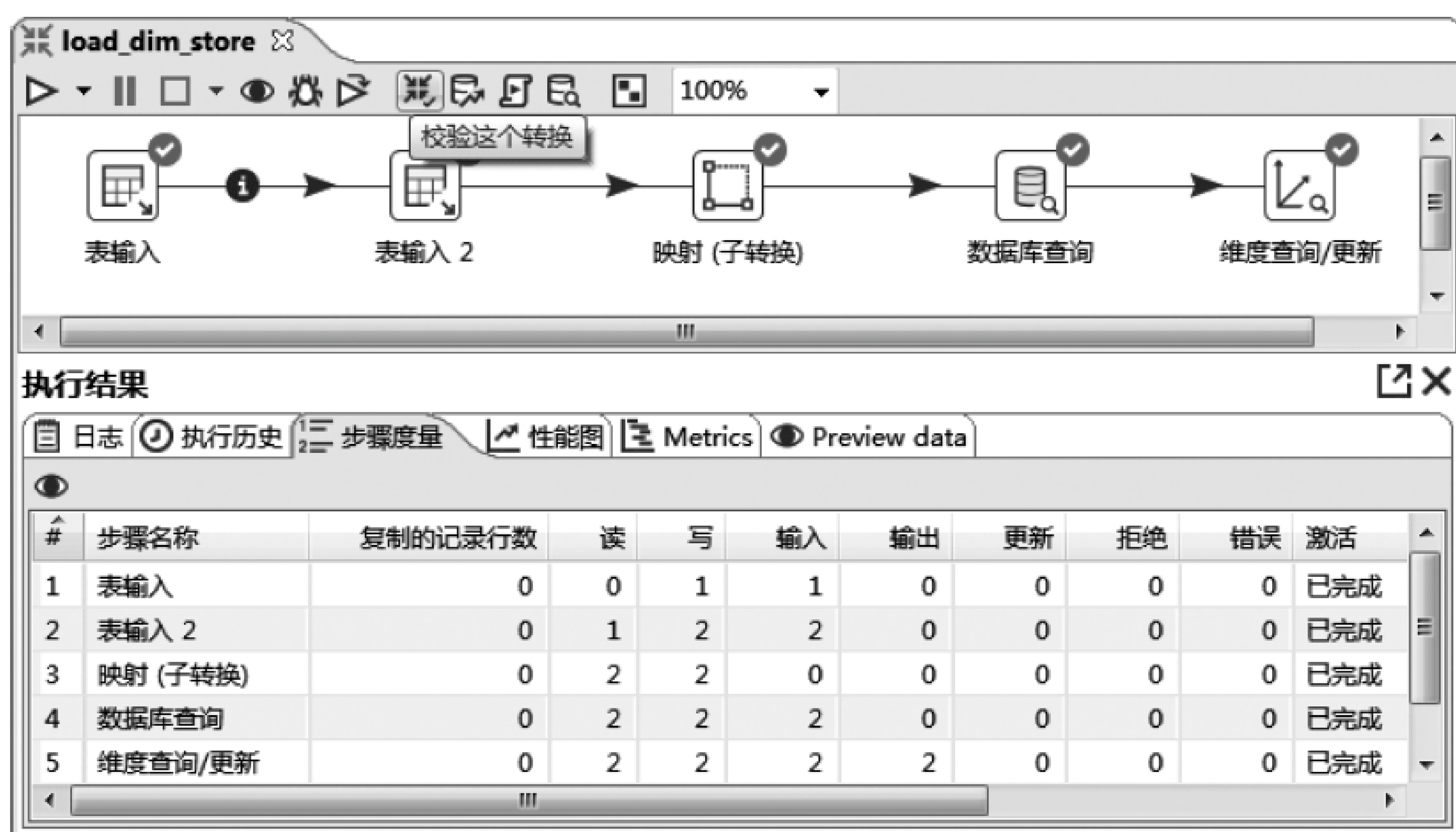


图 8-115　运行转换 load_dim_customer

sto...	store_last_u...	stor...	store_add...	store_di...	store_po...	store_ph...	store_city
1	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
4	2006-02-15 04:57	1	47 MySakila D	Alberta	(NULL)	(NULL)	Lethbridge
5	2006-02-15 04:57	2	28 MySQL Boul	QLD	(NULL)	(NULL)	Woodridge

图 8-116　维度表 dim_store

从图 8-116 中可以看出，维度表 dim_store 中已插入数据，说明我们成功实现了加载商店数据至商店维度表 dim_store。

8.3.7　加载演员数据至演员维度表

下面通过 Kettle 工具加载演员数据至演员维度表 dim_actor，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_actor，并添加“表输入”控件、“插入/更新”控件以及 Hop 跳连接线，具体效果如图 8-117 所示。

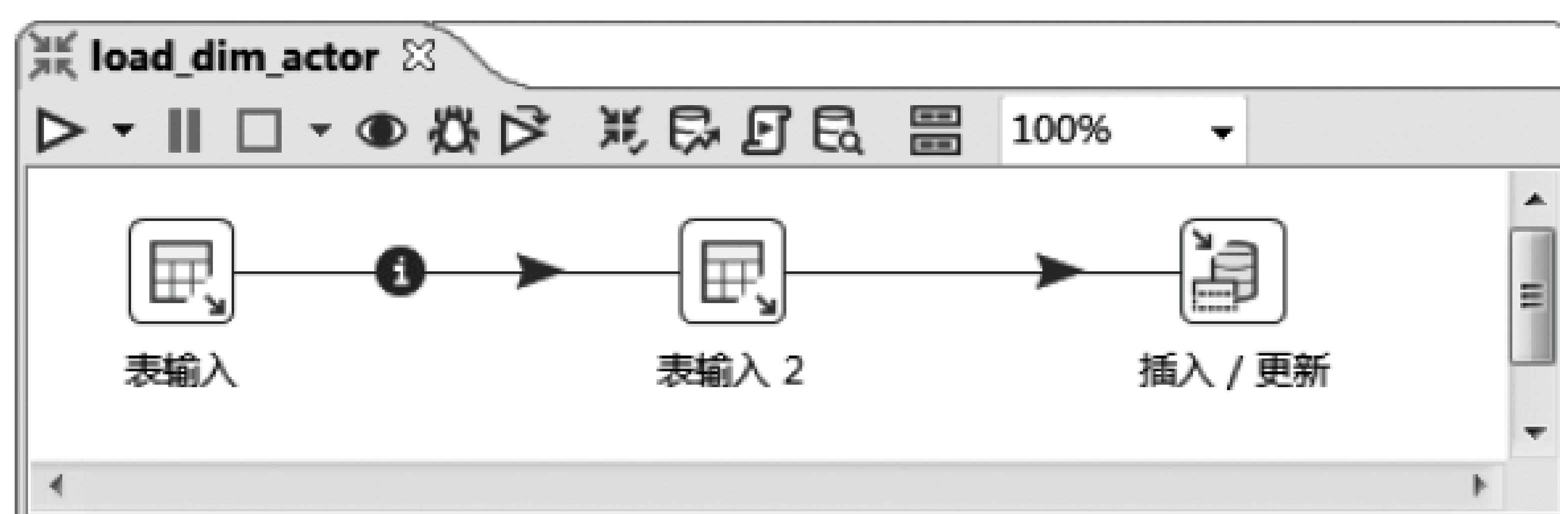


图 8-117　创建转换 load_dim_actor

2. 配置“表输入”控件

双击图 8-117 中的“表输入”控件，进入“表输入”界面，如图 8-118 所示。

在图 8-118 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。



图 8-118 “表输入”界面

MySQL 数据库连接的配置如图 8-119 所示。



图 8-119 MySQL 数据库连接的配置

在图 8-118 的 SQL 框中编写 SQL 语句,用于获取字段 actor_last_update 中的最大值,将该值替换为 1970-01-01 00:00:00 并赋值给临时字段 max_dim_actor_last_update;单击“预览”按钮,查看临时字段 max_dim_actor_last_update 是否将默认值设置为 1970-01-01 00:00:00,具体如图 8-120 和图 8-121 所示。

从图 8-121 中可以看出,临时字段 max_dim_actor_last_update 的默认值设置为 1970-01-01 00:00:00,单击“关闭”→“确定”按钮,完成“表输入”控件的配置。



图 8-120　编写 SQL 语句



图 8-121　预览数据

3. 配置“表输入 2”控件

双击图 8-117 中的“表输入 2”控件，进入“表输入”界面，如图 8-122 所示。



图 8-122　“表输入”界面

在图 8-122 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-123 所示。

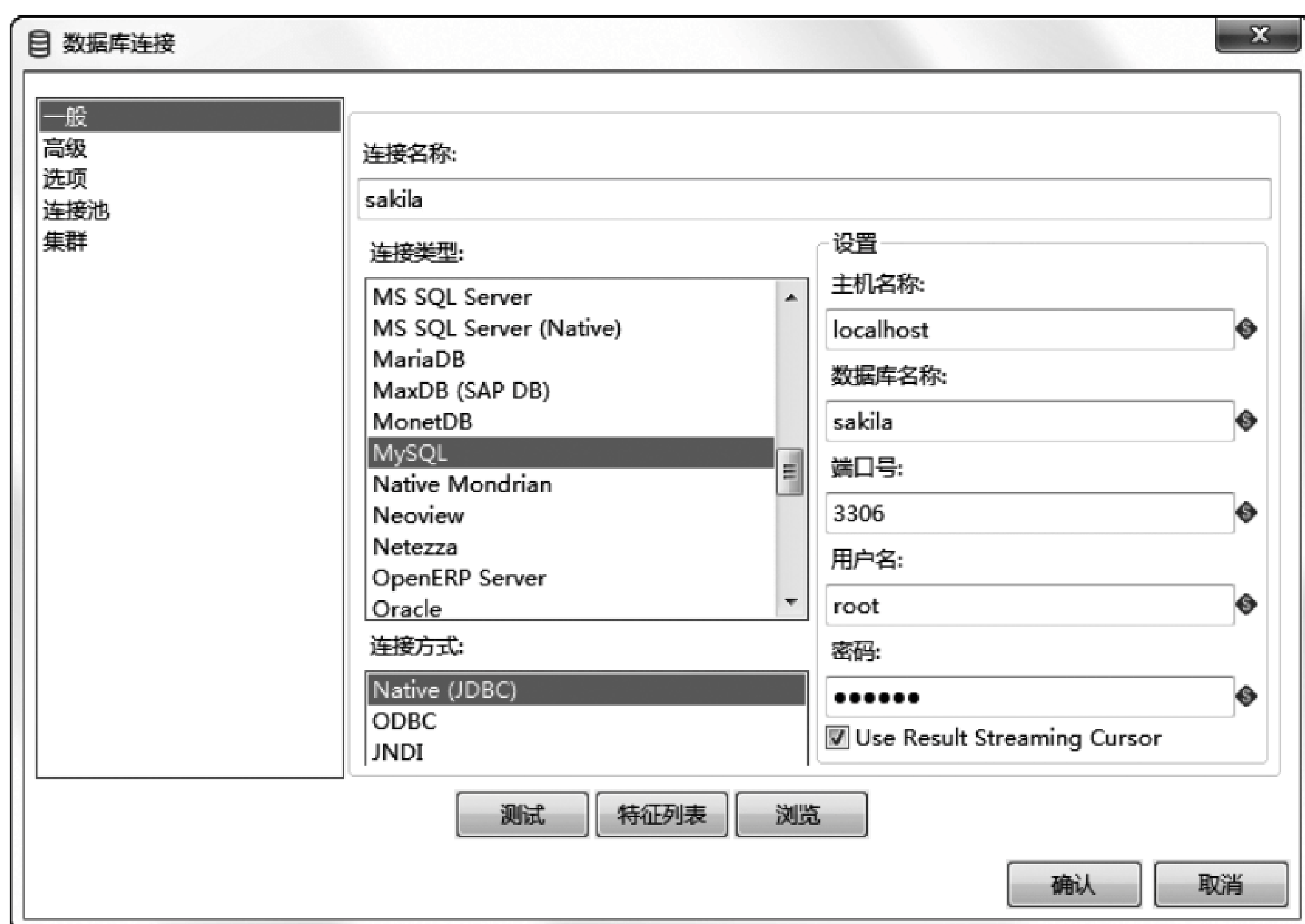


图 8-123 MySQL 数据库连接的配置

在图 8-122 的 SQL 框中编写 SQL 语句,用于获取数据库 sakila 中数据表 actor 中的最新数据,具体如图 8-124 所示。



图 8-124 编写 SQL 语句

在图 8-124 中单击“确定”按钮,完成“表输入 2”控件的配置。

4. 配置“插入/更新”控件

双击图 8-117 中的“插入/更新”控件,进入“插入/更新”界面,如图 8-125 所示。

在图 8-125 中单击“新建”按钮,配置数据库连接,配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-126 所示。



图 8-125　“插入/更新”界面



图 8-126　MySQL 数据库连接的配置

单击图 8-125 中目标表右侧的“浏览”按钮，弹出“数据库浏览器”窗口，选择目标表，即维度表 dim_actor，具体如图 8-127 所示。

在图 8-127 中单击“获取字段”按钮，指定查询数据需要的关键字，这里选择的是维度表 dim_actor 中的 actor_id 字段和输入流里的 actor_id 字段；单击“获取和更新字段”按钮，指定需要更新的字段，具体如图 8-128 所示。

在图 8-128 中单击“确定”按钮，完成“插入/更新”控件的配置。



图 8-127　选择要插入数据的目标表 dim_actor

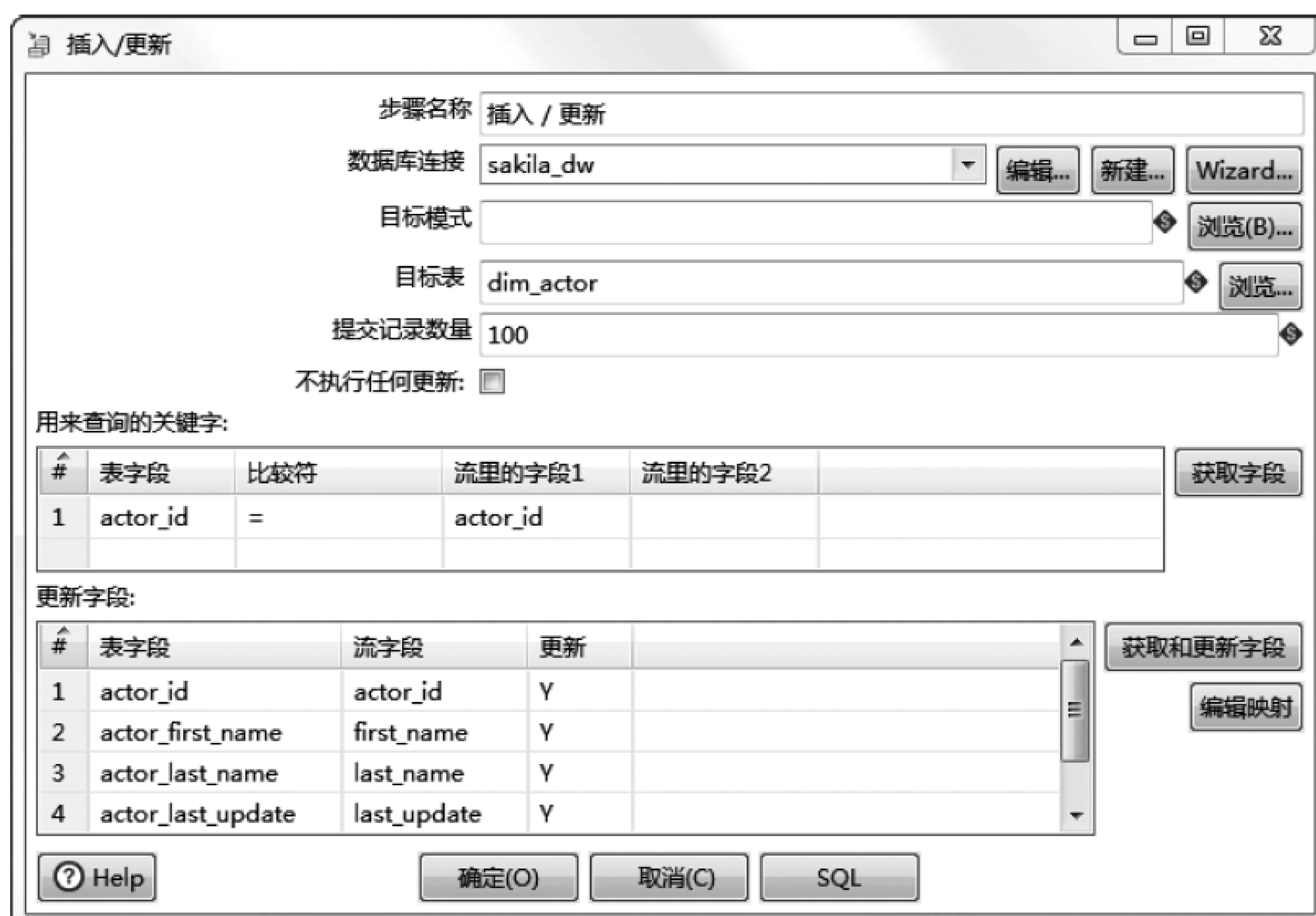


图 8-128　配置"插入/更新"控件

5. 运行转换 load_dim_actor

单击转换工作区顶部的▷按钮,运行创建的转换 load_dim_actor,实现加载演员数据至演员维度表 dim_actor,具体如图 8-129 所示。

从图 8-129 中执行结果的"步骤度量"可以看出,"表输入"控件输入 1 条数据并写入该控件;"表输入 2"控件输入 200 条数据,并从"表输入"控件中读取 1 条数据,写入该控件 200 条数据;"插入/更新"控件输入 200 条数据,并从"表输入 2"控件中读取 200 条数据,写入该控件 200 条数据,最终进行输出。

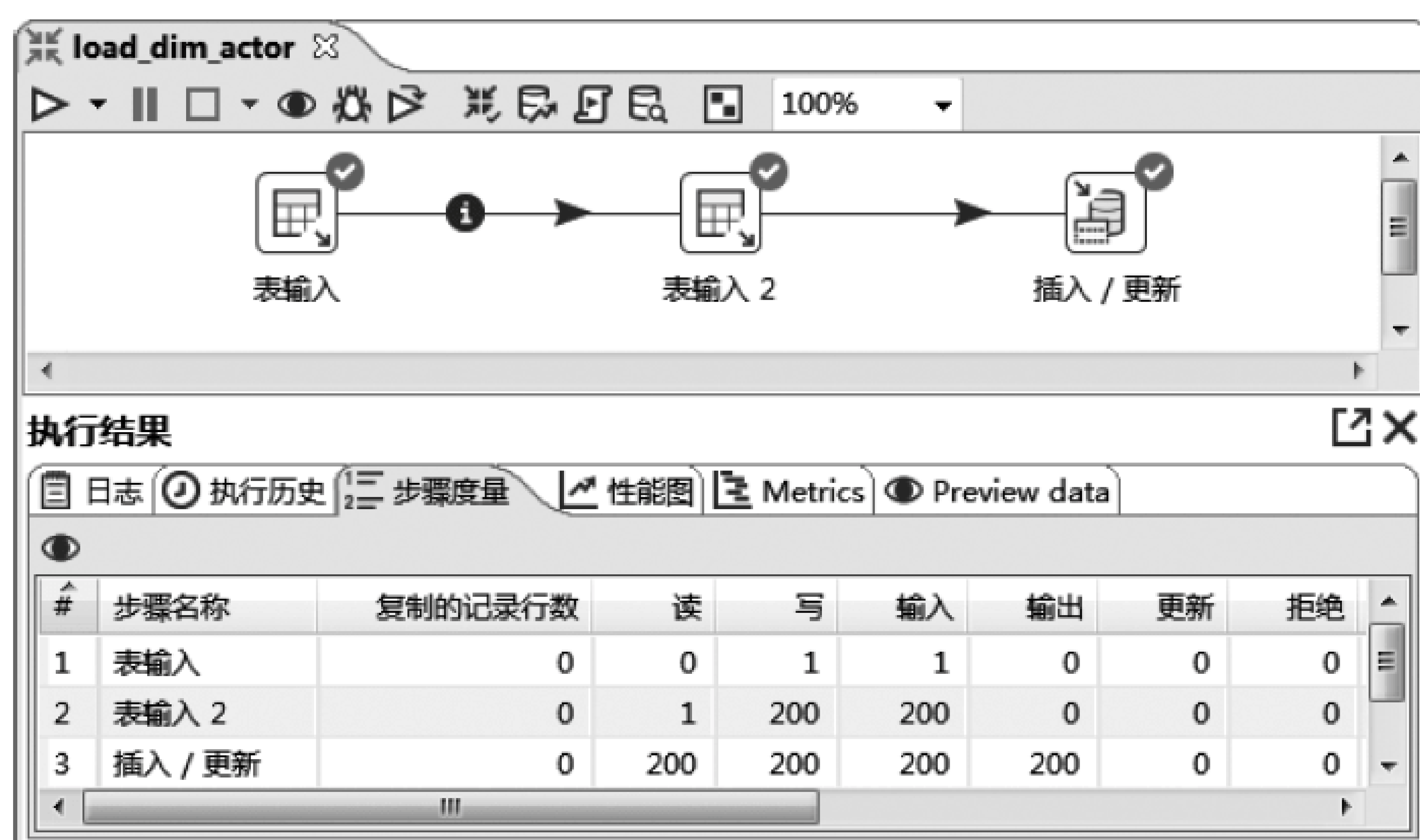


图 8-129　运行转换 load_dim_actor

6. 查看维度表 dim_actor 中的数据

通过 SQLyog 工具，查看维度表 dim_actor 是否已成功插入演员数据，查看结果如图 8-130 所示。

actor_key	actor_last_update	actor_last_name	actor_first_name	actor_id
201	2006-02-15 04:34:33	GUINESS	PENELOPE	1
202	2006-02-15 04:34:33	WAHLBERG	NICK	2
203	2006-02-15 04:34:33	CHASE	ED	3
204	2006-02-15 04:34:33	DAVIS	JENNIFER	4
205	2006-02-15 04:34:33	LOLLOBRIGIDA	JOHNNY	5
206	2006-02-15 04:34:33	NICHOLSON	BETTE	6
207	2006-02-15 04:34:33	MOSTEL	GRACE	7

图 8-130　维度表 dim_actor

从图 8-130 中可以看出，维度表 dim_actor 中已插入数据，说明我们成功实现了加载演员数据至演员维度表 dim_actor。

8.3.8　加载电影数据至电影维度表

下面通过 Kettle 工具加载电影数据至电影维度表 dim_film，具体实现步骤如下。

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_dim_film，并添加“表输入”控件、“数据库查询”控件、“值映射”控件、“列拆分为多行”控件、“增加常量”控件、“列转行”控件、“计算器”控件、“数据库连接”控件、“联合查询/更新”控件、“分组”控件、“流查询”控件、“插入/更新”控件以及 Hop 跳连接线，具体效果如图 8-131 所示。

2. 配置“表输入”控件

双击图 8-131 中的“表输入”控件，进入“表输入”界面，如图 8-132 所示。

在图 8-132 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。

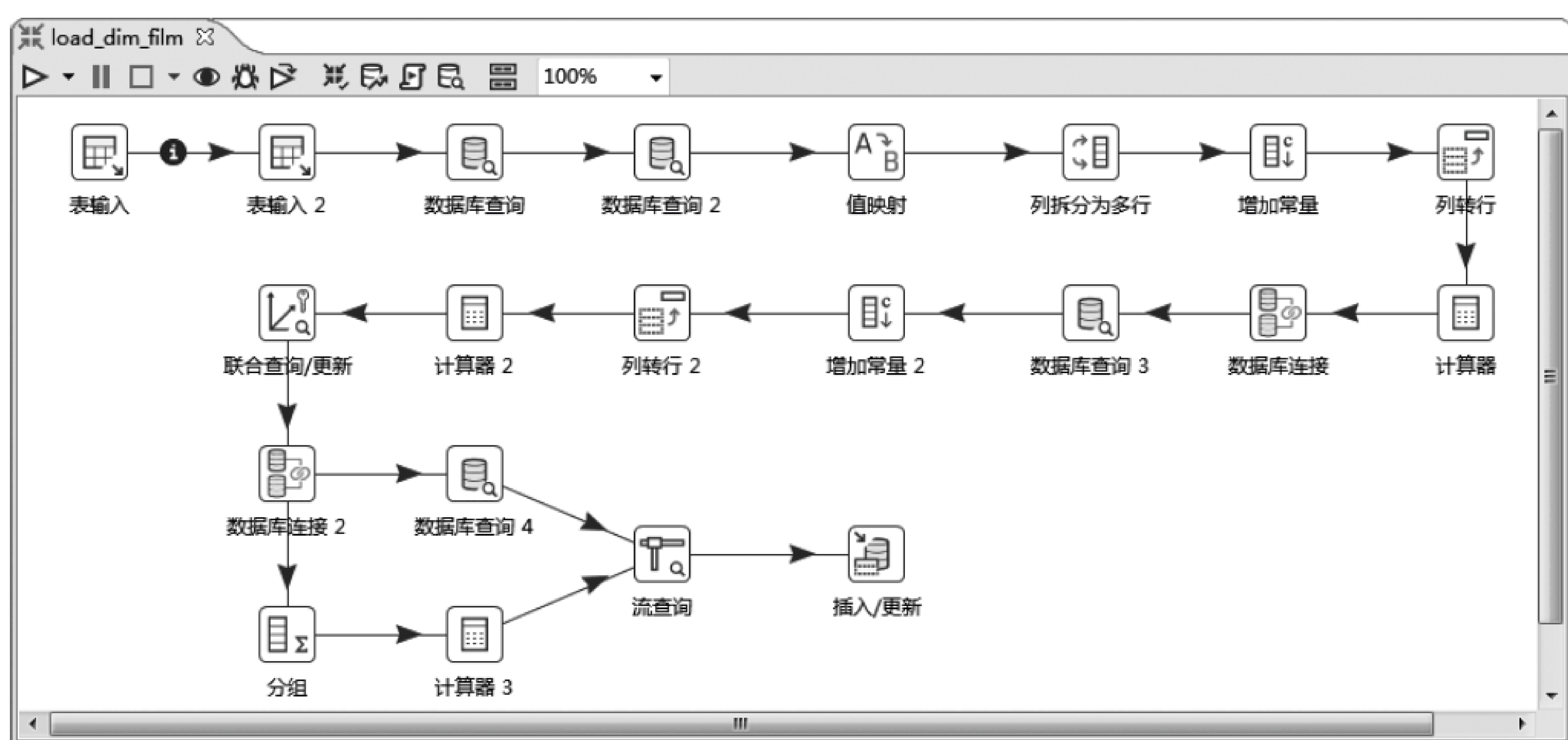


图 8-131 创建转换 load_dim_film



图 8-132 “表输入”界面

MySQL 数据库连接的配置如图 8-133 所示。

在图 8-132 的 SQL 框中编写 SQL 语句，用于获取字段 film_last_update 中的最大值，将该值替换为 1970-01-01 00:00:00 并赋值给临时字段 max_dim_film_last_update；单击“预览”按钮，查看临时字段 max_dim_film_last_update 是否将默认值设置为 1970-01-01 00:00:00”，具体如图 8-134 和图 8-135 所示。

从图 8-135 中可以看出，临时字段 max_dim_film_last_update 的默认值设置为 1970-01-01 00:00:00，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。

3. 配置“表输入 2”控件

双击图 8-131 中的“表输入 2”控件，进入“表输入”界面，如图 8-136 所示。

在图 8-136 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-137 所示。



图 8-133　MySQL 数据库连接的配置



图 8-134　编写 SQL 语句



图 8-135　预览数据



图 8-136　“表输入”界面

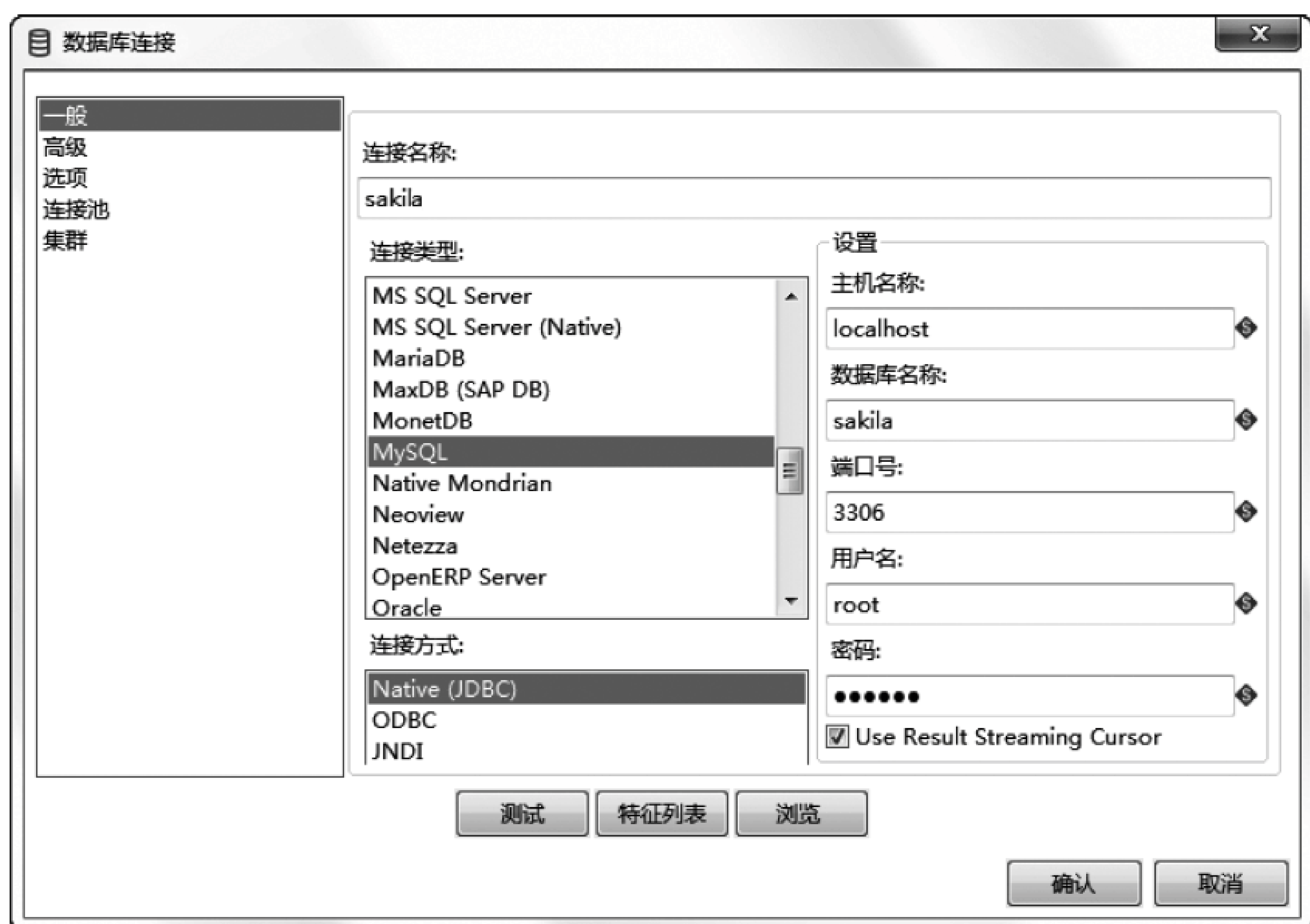


图 8-137　MySQL 数据库连接的配置

在图 8-136 的 SQL 框中编写 SQL 语句，用于获取数据库 sakila 中数据表 film 中的最新数据，具体如图 8-138 所示。

在图 8-138 中单击“确定”按钮，完成“表输入 2”控件的配置。

4. 配置“数据库查询”控件

双击图 8-131 中的“数据库查询”控件，进入“数据库查询”界面，如图 8-139 所示。

在图 8-139 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-140 所示。

单击图 8-139 中表名后的“浏览”按钮，添加数据表 language，用于查询电影的语言，如



图 8-138　编写 SQL 语句



图 8-139　“数据库查询”界面

图 8-141 所示。

在图 8-141 的“查询所需的关键字”框中添加查询所需的关键字字段 language_id，用于指定表字段和流字段的比较条件；在“查询表返回的值”框中添加查询表返回的字段 name，并重命名为 language，如图 8-142 所示。

在图 8-142 中单击“确定”按钮，完成“数据库查询”控件的配置。

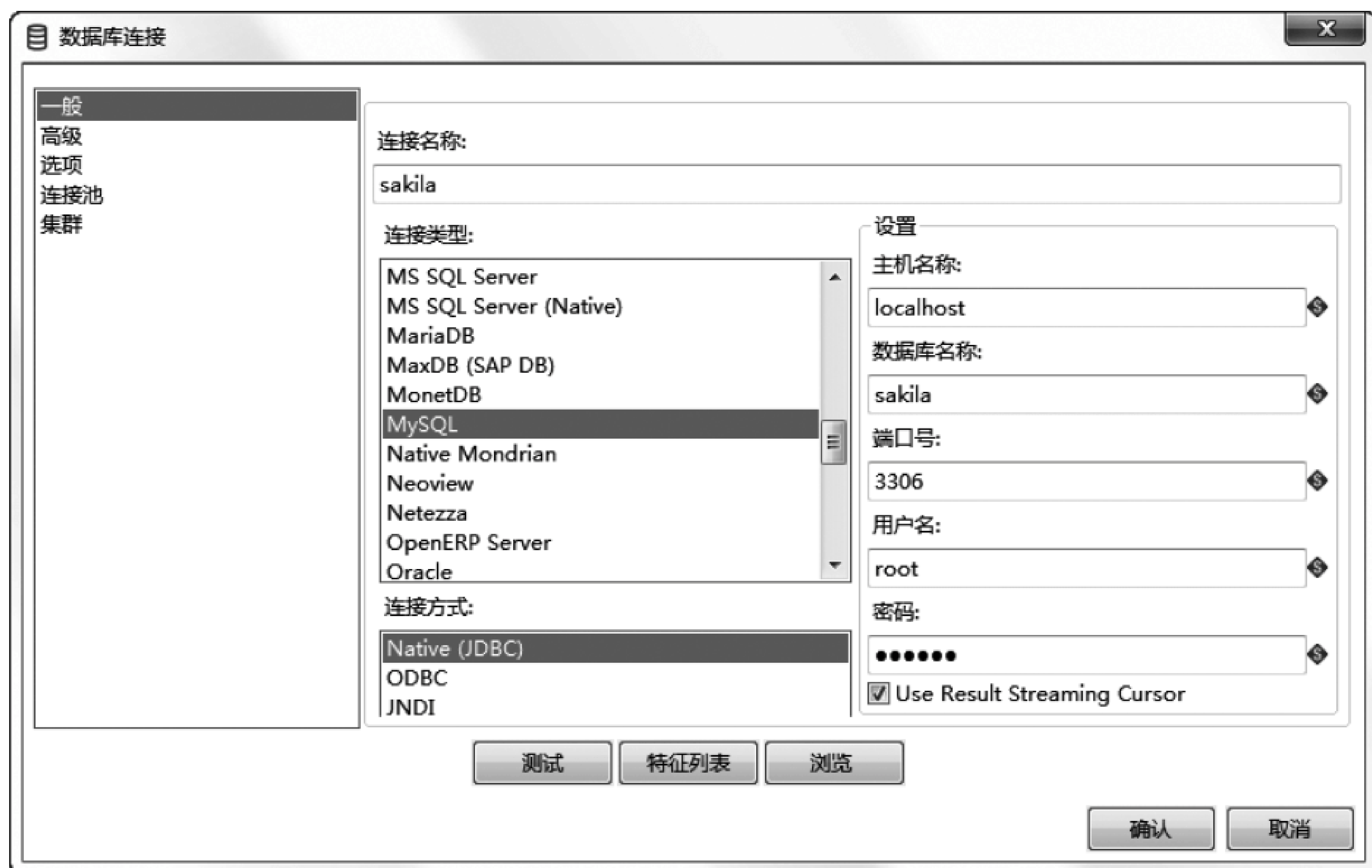


图 8-140 MySQL 数据库连接的配置



图 8-141 添加数据表 language

5. 配置“数据库查询 2”控件

双击图 8-131 中的“数据库查询 2”控件，进入“数据库查询”界面，如图 8-143 所示。

在图 8-143 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-144 所示。



图 8-142　配置“数据库查询”控件



图 8-143　“数据库查询”界面

单击图 8-143 中表名后的“浏览”按钮，添加数据表 language，用于查询电影原版的语言，如图 8-145 所示。

在图 8-145 的“查询所需的关键字”框中添加查询所需的关键字字段 language_id 和 original_language，用于指定表字段和流字段的比较条件；在“查询表返回的值”框中添加查询表返回字段 name 并重命名为 original_language，若对于电影原版语言为 Null 值，则设置



图 8-144 MySQL 数据库连接的配置



图 8-145 添加数据表 language

默认值为 Not Applicable,如图 8-146 所示。

在图 8-146 中单击"确定"按钮,完成"数据库查询 2"控件的配置。

6. 配置"值映射"控件

双击图 8-131 中的"值映射"控件,进入"值映射"界面,在"使用的字段名"后的下拉列表

数据库查询

步骤名称 数据库查询 2

数据库连接 sakila 编辑... 新建... Wizard...

模式名称 浏览(B)...

表名 language 浏览...

使用缓存

缓存大小 0

从表中加载所有数据

查询所需的关键字:

#	表字段	比较操作符	字段1	字段2
1	language_id	=	original_language_id	

查询表返回的值:

#	字段	新的名称	默认	类型
1	name	original_language	Not Applicable	String

查询失败则忽略

多行结果时失败

排序

Help 确定(O) 取消(C) 获取查询关键字 获取返回字段

图 8-146 配置“数据库查询 2”控件

中选择字段 rating,指定对电影级别的字段 rating 进行映射;在“目标字段名(空=覆盖)”处添加 rating_text 字段,用于存储目标值;在“字段值”框中添加源值和目标值,其中,源值为 G(大众级,所有年龄的观众均可观看)、PG(普通级,建议在父母的陪伴下观看)、PG-13(普通级,不适于 13 岁以下儿童,需要父母陪同观看)、R(限制级,17 岁以下观众必须由父母或者监护陪伴才能观看)、NC-17(禁止 17 岁或者 17 岁以下观众观看),这些均为美国电影分级级别的简称,对应的全称目标值为 General Audiences、Parental Guidance Suggested、Parents Strongly Cautioned、Restricted、No One Under 17 Admitted,如图 8-147 所示。

值映射

步骤名称: 值映射

使用的字段名: rating

目标字段名 (空=覆盖): rating_text

不匹配时的默认值: Not Yet Rated

字段值:

#	源值	目标值
1	G	General Audiences
2	PG	Parental Guidance Suggested
3	PG-13	Parents Strongly Cautioned
4	R	Restricted
5	NC-17	No One Under 17 Admitted

Help 确定(O) 取消(C)

图 8-147 配置“值映射”控件

在图 8-147 中单击“确定”按钮,完成“值映射”控件的配置。

7. 配置“列拆分为多行”控件

双击图 8-131 中的“列拆分为多行”控件，进入“列拆分为多行”界面，由于 special_features 字段表示的是电影的特点，而电影的特点有多个，因此要进行拆分；在“要拆分的字段”后的下拉列表中选择要拆分的字段 special_features；在“分隔符”处指定分隔符“,”；在“新字段名”框中添加新的字段名，用于存放利用分隔符分隔后的数据，具体如图 8-148 所示。



图 8-148 配置“列拆分为多行”控件

在图 8-148 中单击“确定”按钮，完成“列拆分为多行”控件的配置。

8. 配置“增加常量”控件

双击图 8-131 中的“增加常量”控件，进入“增加常量”界面，在“字段”框中添加常量字段 Yes 和 No，并指定值为 Yes 和 No，用于后续判断某电影是否有预告片、是否有评论、是否删减片段以及是否有幕后等内容，若有，则用 Yes 标识，反之用 No 标识，如图 8-149 所示。



图 8-149 配置“增加常量”控件

在图 8-149 中单击“确定”按钮，完成“增加常量”控件的配置。

9. 配置“列转行”控件

双击图 8-131 中的“列转行”控件，进入“列转行”界面，在“关键字段”后的下拉列表中选择关键字段 special_feature，由于字段 special_feature 中包含一个或多个内容，因此需要对字段 special_feature 进行列转行操作；在“构成分组的字段”框中添加分组字段；在“目标字

段”框中添加目标字段，具体如图 8-150 所示。

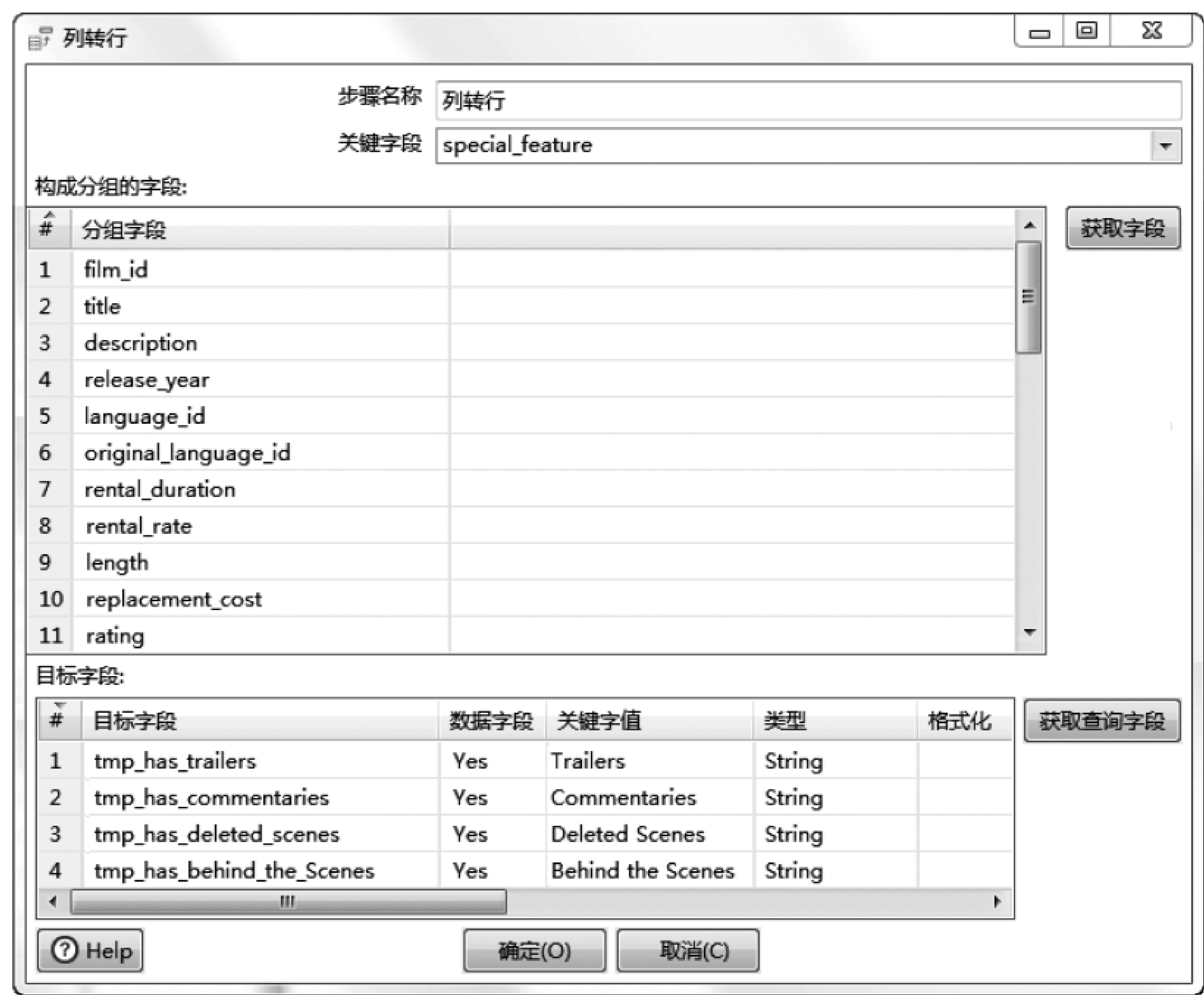


图 8-150　配置“列转行”控件

在图 8-150 中单击“确定”按钮，完成“列转行”控件的配置。

10．配置“计算器”控件

双击图 8-131 中的“计算器”控件，进入“计算器”界面，在“字段”处添加新字段，用于存储将“列转行”控件流中的字段 special_feature 的 NULL 值替换成 No 的数据，具体如图 8-151 所示。

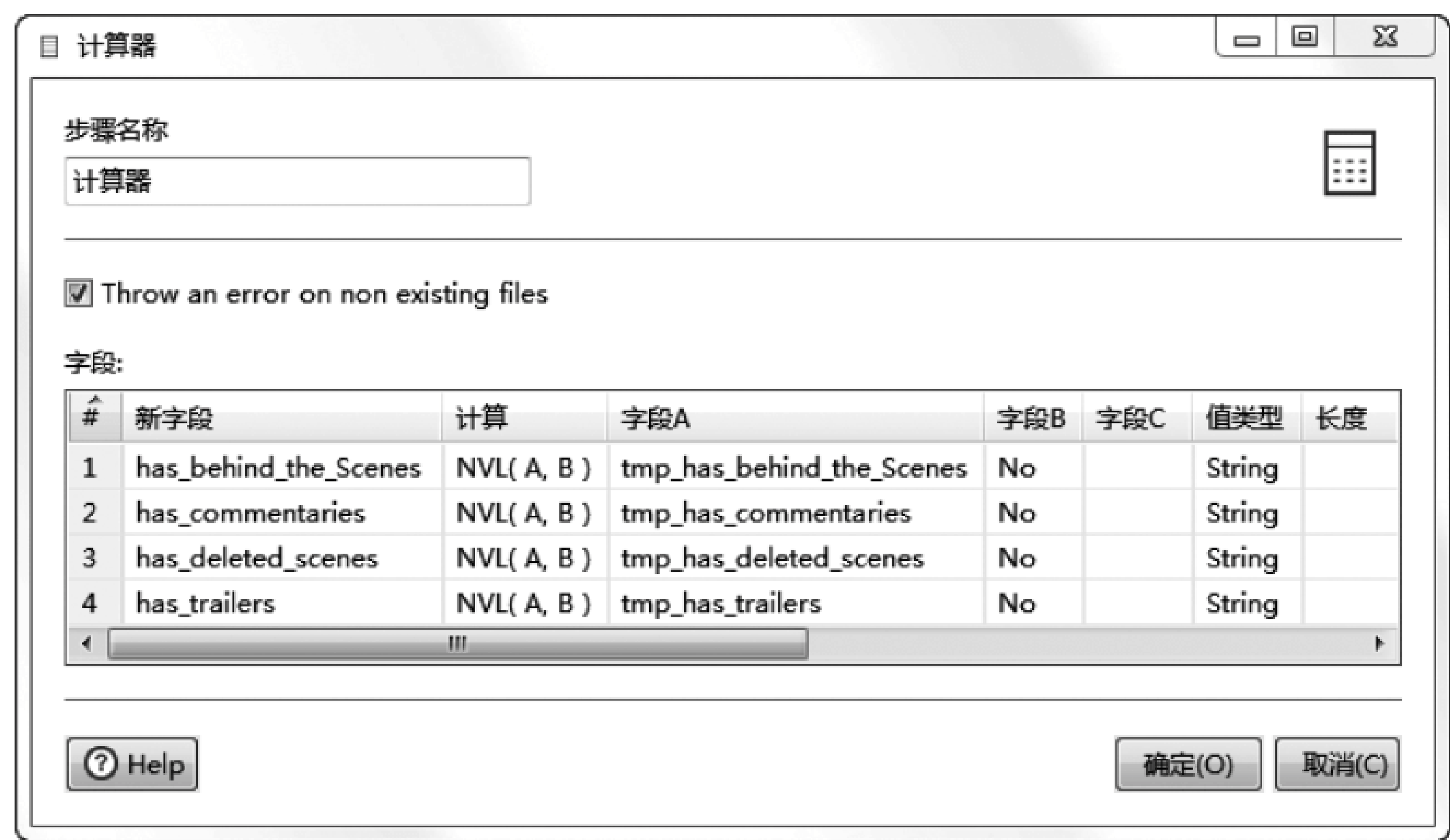


图 8-151　配置“计算器”控件

在图 8-151 中，NVL(A,B)的含义是实现将字段 A 中的空值替换成字段 B(即 No)，单击“确定”按钮，完成“计算器”控件的配置。

11．配置“数据库连接”控件

双击图 8-131 中的“数据库连接”控件，进入 Database join 界面，如图 8-152 所示。



图 8-152 Database join 界面

在图 8-152 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-153 所示。

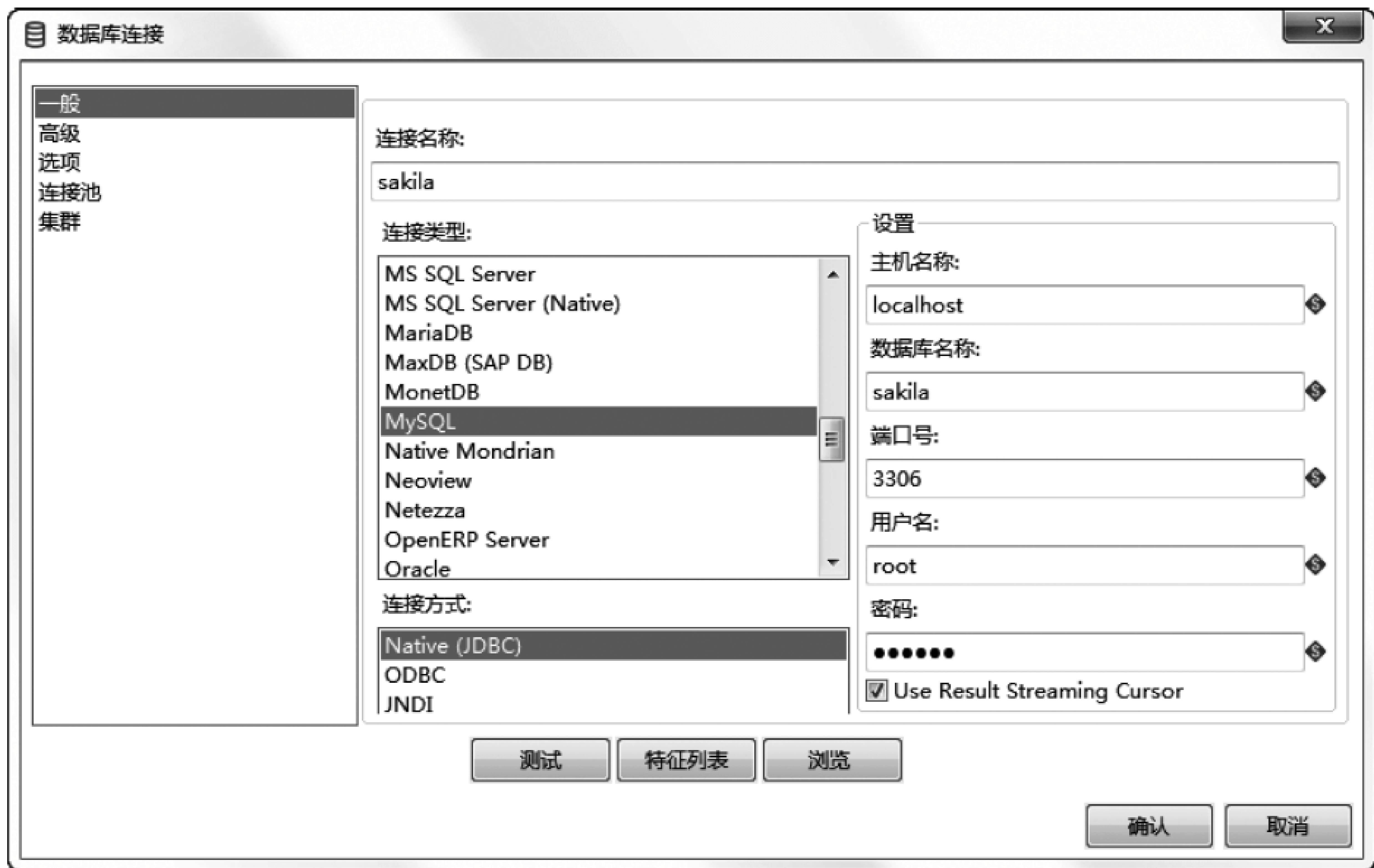


图 8-153 MySQL 数据库连接的配置

在图 8-152 的 SQL 框中编写 SQL 语句，用于根据电影 id 获取电影的分类；在 The

parameters to use 框中添加编写 SQL 语句需要的参数，即 film_id，“计算器”控件流中的字段 film_id 作为参数进行传递，供“数据库连接”控件使用，具体如图 8-154 所示。



图 8-154　配置“数据库连接”控件

在图 8-154 中单击“确定”按钮，完成“数据库连接”控件的配置。

12. 配置“数据库查询 3”控件

双击图 8-131 中的“数据库查询 3”控件，进入“数据库查询”界面，如图 8-155 所示。



图 8-155　“数据库查询”界面

在图 8-155 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-156 所示。



图 8-156 MySQL 数据库连接的配置

单击图 8-155 中表名后的“浏览”按钮，添加数据表 category，用于获取电影分类的 id；在“查询所需的关键字”框中添加查询所需的关键字字段 category_id，用于指定表字段和流字段的比较条件；在“查询表返回的值”框中添加查询表返回的值，即 name，并重命名为 category，如图 8-157 所示。



图 8-157 配置“数据库查询 3”控件

在图 8-157 中单击“确定”按钮，完成“数据库查询 3”控件的配置。

13. 配置“增加常量 2”控件

双击图 8-131 中的“增加常量 2”控件，进入“增加常量”界面，在“字段”框中添加常量字段 Yes，由于在“增加常量”控件中设置的常量 Yes 已经被替换，所以需要添加一个常量，具体如图 8-158 所示。



图 8-158　配置“增加常量 2”控件

在图 8-158 中单击“确定”按钮，完成“增加常量 2”控件的配置。

14. 配置“列转行 2”控件

双击图 8-131 中的“列转行 2”控件，进入“列转行”界面，在“关键字段”后的下拉列表中选择关键字段 category，根据电影分类的名称对电影进行分类；在“构成分组的字段”框中添加分组字段，如图 8-159 所示；在“目标字段”框中添加目标字段，具体如图 8-160 所示。

在图 8-160 中单击“确定”按钮，完成“列转行”控件的配置。

15. 配置“计算器 2”控件

双击图 8-131 中的“计算器 2”控件，进入“计算器”界面，在“字段”处添加新字段，用于将“列转行 2”控件流中字段 category 中的 NULL 值替换成 No，具体如图 8-161 所示。

在图 8-161 中单击“确定”按钮，完成“计算器 2”控件的配置。

16. 配置“联合查询/更新”控件

双击图 8-131 中的“联合查询/更新”控件，进入“联合查询/更新”界面，如图 8-162 所示。

在图 8-162 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-163 所示。

单击图 8-162 中目标表右侧的“浏览”按钮，选择目标表，即维度表 dim_film，用于加载最新的电影维度数据，在“代理关键字”框中添加代理关键字段，并指定创建代理键为自增字段，如图 8-164 所示；在“关键字段”框中添加维度字段和流里的字段，具体如图 8-165 所示。

单击图 8-165 中的“确定”按钮，完成“联合查询/更新”控件的配置。

17. 配置“数据库连接 2”控件

双击图 8-131 中的“数据库连接 2”控件，进入 Database join 界面，如图 8-166 所示。

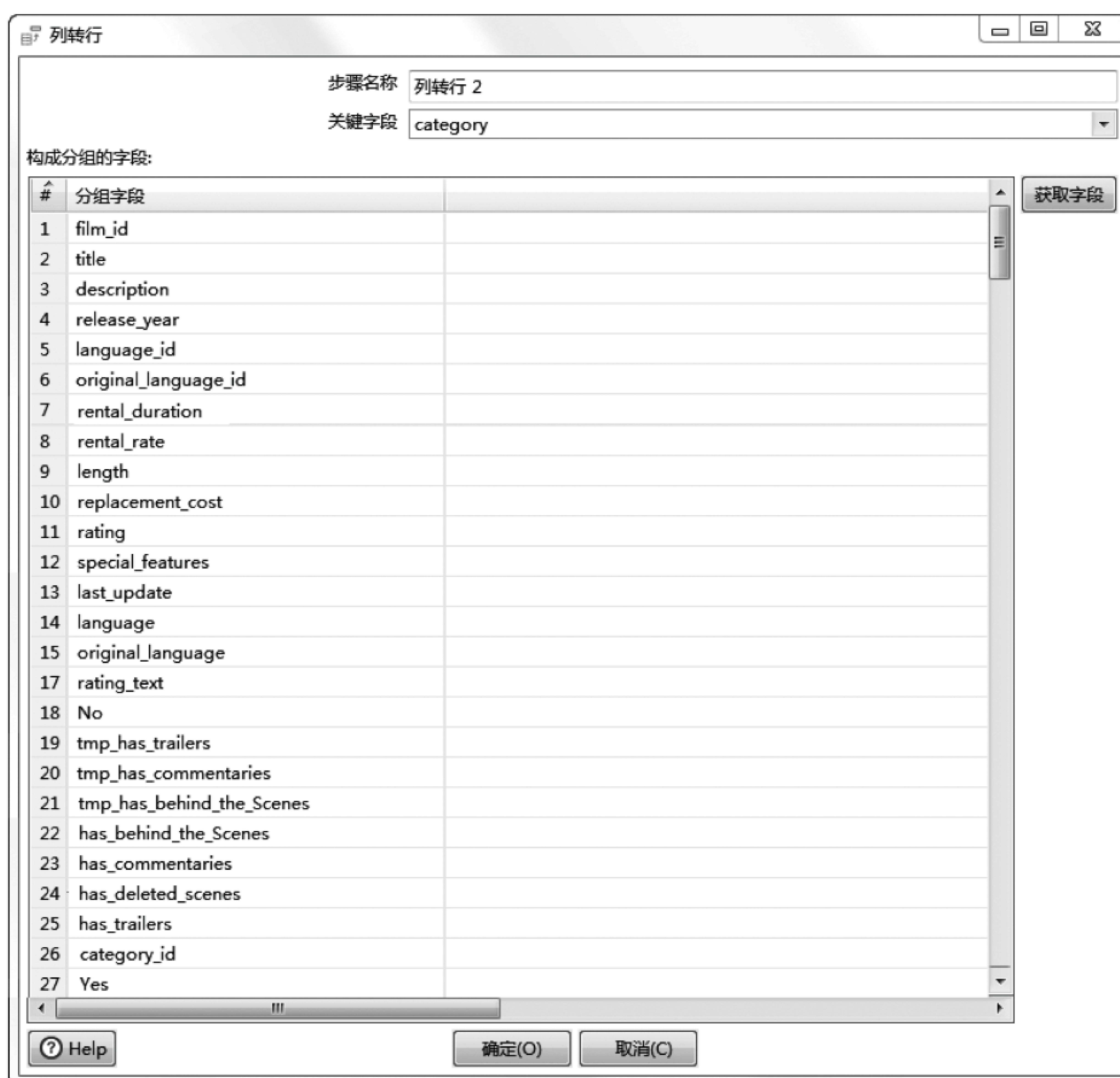


图 8-159　添加关键字段和构成分组的字段



图 8-160　添加目标字段

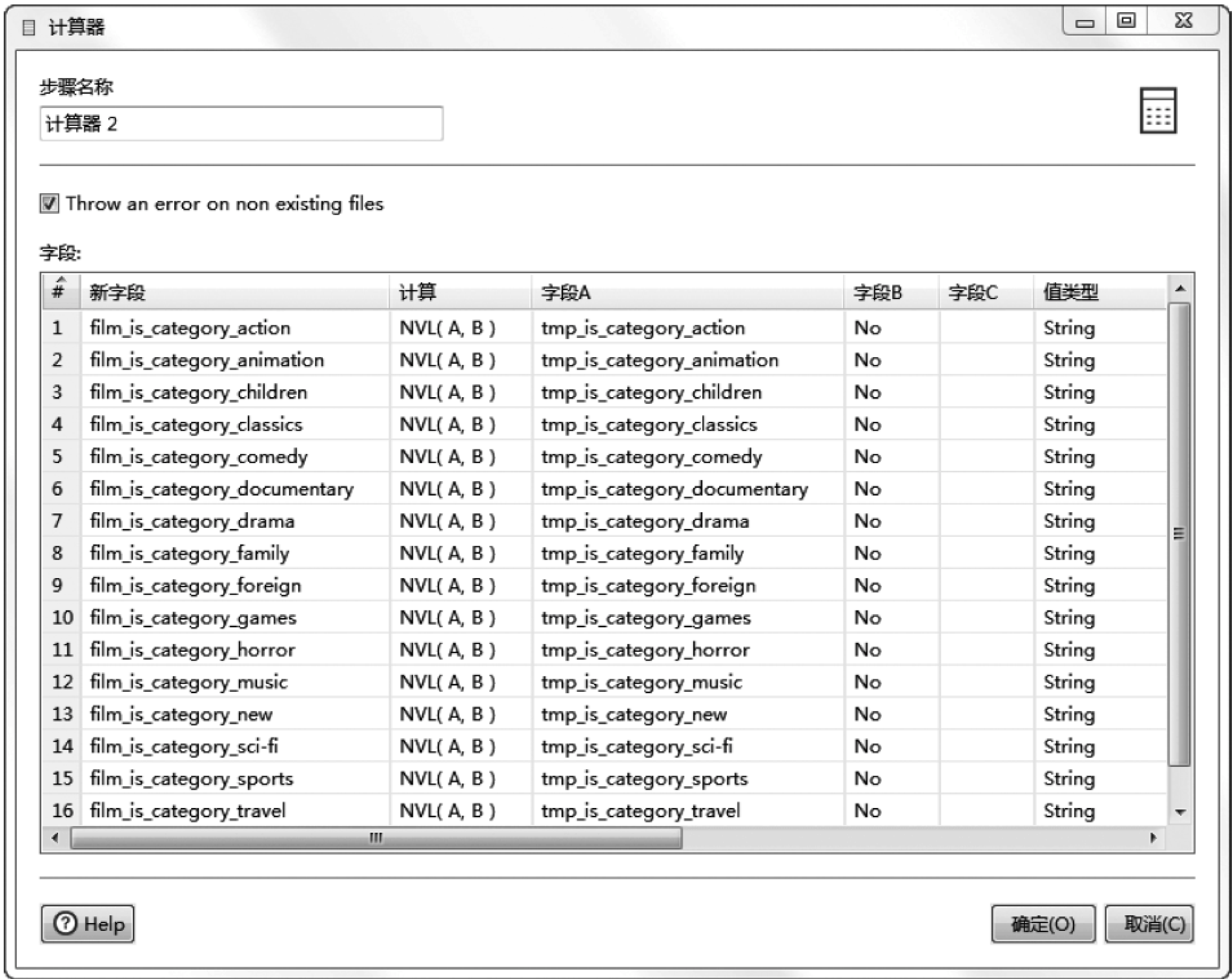


图 8-161　配置“计算器 2”控件



图 8-162　“联合查询/更新”界面



图 8-163 MySQL 数据库连接的配置



图 8-164 添加目标表、代理关键字

#	维度字段	流里的字段
1	film_id	film_id
2	film_title	title
3	film_description	description
4	film_release_year	release_year
5	film_rental_duration	rental_duration
6	film_rental_rate	rental_rate
7	film_replacement_cost	replacement_cost
8	film_rating_code	rating
9	film_last_update	last_update
10	film_language	language
11	film_original_language	original_language
12	film_rating_text	rating_text
13	film_has_behind_the_scenes	has_behind_the_Scenes
14	film_has_commentaries	has_commentaries
15	film_has_deleted_scenes	has_deleted_scenes
16	film_has_trailers	has_trailers
17	film_in_category_action	film_is_category_action
18	film_in_category_animation	film_is_category_animation
19	film_in_category_children	film_is_category_children
20	film_in_category_classics	film_is_category_classics
21	film_in_category_comedy	film_is_category_comedy
22	film_in_category_documentary	film_is_category_documentary
23	film_in_category_drama	film_is_category_drama
24	film_in_category_family	film_is_category_family
25	film_in_category_foreign	film_is_category_foreign
26	film_in_category_games	film_is_category_games
27	film_in_category_horror	film_is_category_horror
28	film_in_category_music	film_is_category_music
29	film_in_category_new	film_is_category_new
30	film_in_category_scifi	film_is_category_sci-fi
31	film_in_category_sports	film_is_category_sports
32	film_in_category_travel	film_is_category_travel

图 8-165 配置关键字段

Database join

步骤名称 数据库连接 2

数据库连接 编辑... 新建... Wizard...

SQL

Line 1 Column 0

Number of rows to return 0

Outer join?

Replace variables

The parameters to use:

#	Parameter fieldname	Parameter Type
1		

Help 确定(O) 取消(C) 获取字段

图 8-166 Database join 界面

在图 8-166 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-167 所示。



图 8-167 MySQL 数据库连接的配置

在图 8-166 的 SQL 框中编写 SQL 语句，用于根据电影 id 获取演员的 id，从而获取演员的基本信息；在 The parameters to use 框中添加需要的参数，即 film_id，具体如图 8-168 所示。



图 8-168 配置“数据库连接 2”控件

在图 8-168 中单击“确定”按钮，完成“数据库连接 2”控件的配置。

18. 配置"数据库查询 4"控件

双击图 8-131 中的"数据库查询 4"控件，进入"数据库查询"界面，如图 8-169 所示。



图 8-169　"数据库查询"界面

在图 8-169 中单击"新建"按钮，配置数据库连接，配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 8-170 所示。



图 8-170　MySQL 数据库连接的配置

单击图 8-169 中表名后的“浏览”按钮，添加维度表 dim_actor，用于获取演员的基本信息；在“查询所需的关键字”框中添加查询所需的关键字字段 actor_id，用于指定表字段和流字段的比较条件；在“查询表返回的值”框中添加查询表返回的值，如图 8-171 所示。

图 8-171 配置“数据库查询 4”控件

在图 8-171 中单击“确定”按钮，完成“数据库查询 4”控件的配置。

19. 配置“分组”控件

双击图 8-131 中的“分组”控件，进入“分组”界面，如图 8-172 所示。

图 8-172 “分组”界面

在图 8-172 的“构成分组的字段”框中添加分组字段 film_key，对“数据库查询 4”控件流

中的数据按字段 film_key 进行分组；在“聚合”框中添加聚合字段 count_actors，用于统计出演某部电影的演员个数，如图 8-173 所示。



图 8-173　配置“分组”控件

在图 8-173 中单击“确定”按钮，完成“分组”控件的配置。

20. 配置“计算器 3”控件

双击图 8-131 中的“计算器 3”控件，进入“计算器”界面，在“字段”处添加新字段 one 和 actor_weighting_factor，其中字段 one 为自定义的常量，指定值为 1，是一个临时值，可移除；字段 actor_weighting_factor 用于存储演员的权重因子（由于数据库 sakila 中没有可以确定每个演员对电影的实际贡献值，因此可假设每个演员的贡献值相同，因每个演员的权重因子也都相同），具体如图 8-174 所示。

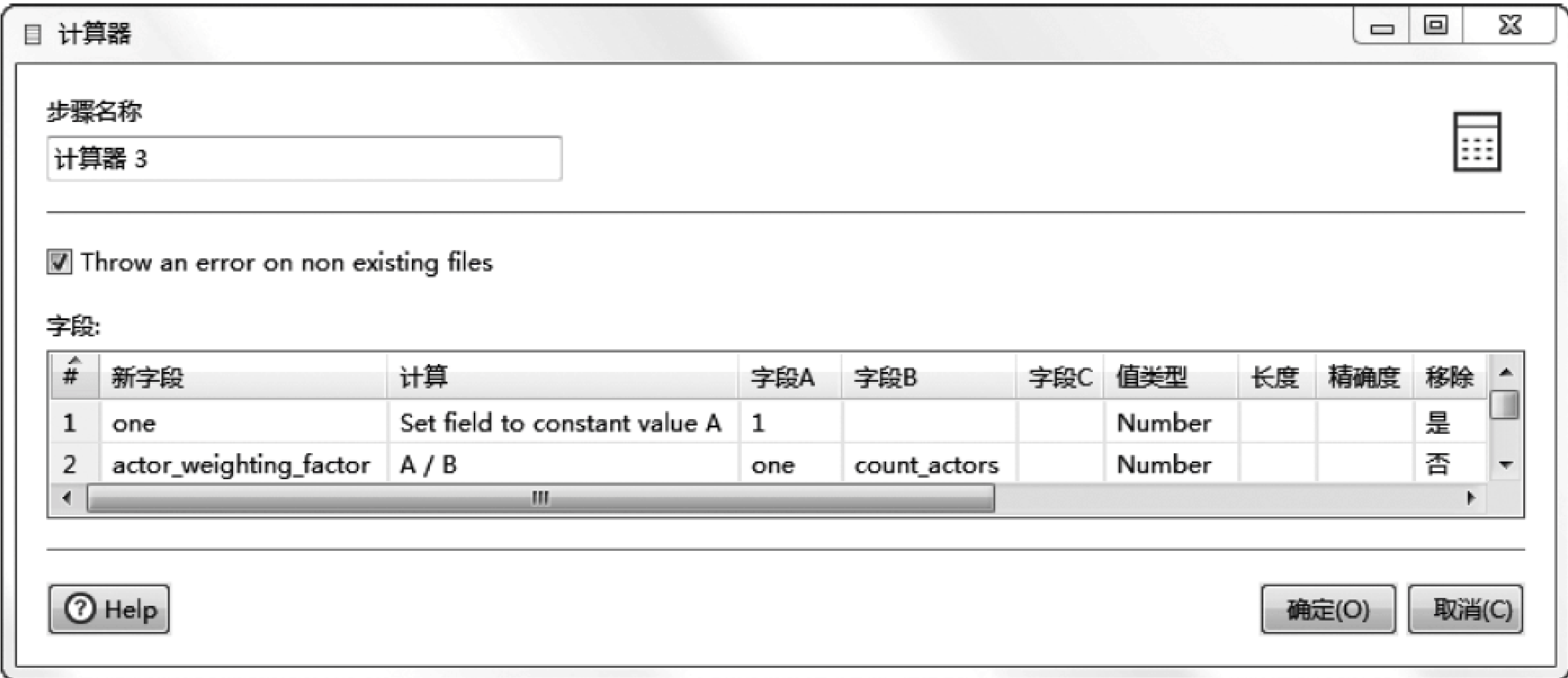


图 8-174　配置“计算器 3”控件

在图 8-174 中单击“确定”按钮，完成“计算器 3”控件的配置。

21. 配置“流查询”控件

双击图 8-131 中的“流查询”控件，进入“流里的值查询”界面，在 Lookup step 后的下拉列表中选择要查询的流，即“计算器 3”控件流；在“查询值所需的关键字”处添加用于查询流里值的字段，即 film_key，通过使用“数据库查询 4”控件中的字段 film_key 去“计算器 3”控件流中匹配相应的数据；在“指定用来接收的字段”框中添加用来接收值的字段 actor_weighting_factor，如图 8-175 所示。

图 8-175 配置“流查询”控件

在图 8-175 中单击“确定”按钮，完成“流查询”控件的配置。

22. 配置“插入/更新”控件

双击图 8-131 中的“插入/更新”控件，进入“插入/更新”界面，如图 8-176 所示。

图 8-176 “插入/更新”界面

在图 8-176 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-177 所示。



图 8-177　MySQL 数据库连接的配置

单击图 8-176 中目标表右侧的“浏览”按钮，弹出“数据库浏览器”窗口，选择目标表，即维度表 dim_film_actor_bridge；单击“获取字段”按钮，用来指定查询数据需要的关键字；单击“获取和更新字段”按钮，用来指定需要更新的字段。“插入/更新”控件用于通过比较流与表中字段的数据，更新维度表 dim_film_actor_bridge 中的数据，如图 8-178 所示。



图 8-178　配置“插入/更新”控件

在图 8-178 中单击“确定”按钮，完成“插入/更新”控件的配置。

23. 运行转换 load_dim_film

单击转换工作区顶部的▷按钮，运行创建的转换 load_dim_film，实现加载电影数据至电影维度表 dim_film，具体如图 8-179 所示。

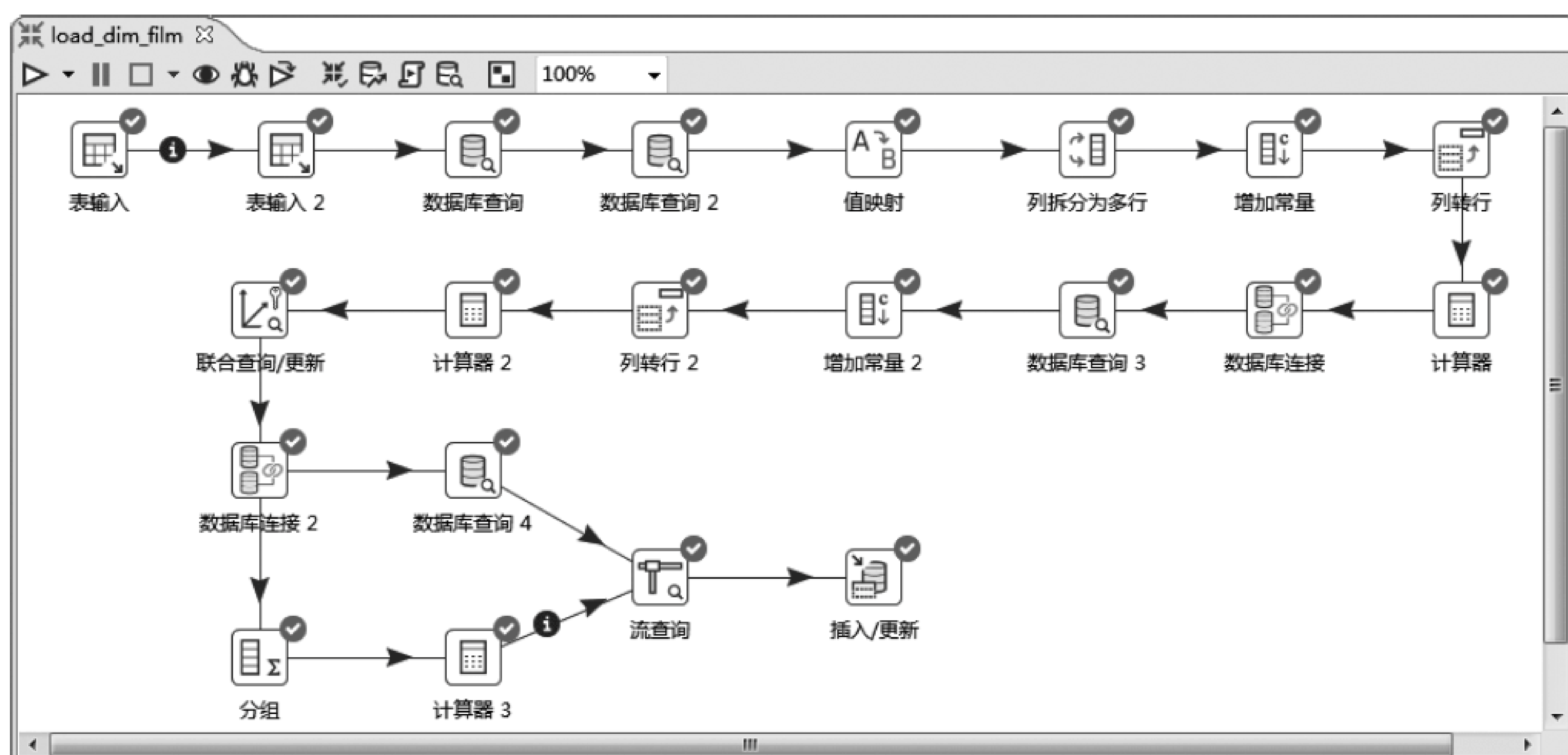


图 8-179　运行转换 load_dim_film

从图 8-179 中的执行结果看，每个控件的右上角均有"√"，说明转换 load_dim_film 执行成功。

24. 查看维度表 dim_film 和 dim_film_actor_bridge 中的数据

通过 SQLyog 工具，查看维度表 dim_film 和 dim_film_actor_bridge 是否已成功插入数据，查看结果如图 8-180 和图 8-181 所示(这里只展示部分数据)。

film_key	film_last_update	film_title	film_description	fil...	film...	film_origin...	fi...	film_...
1001	2006-02-15 05:03:42	ACADEMY DINOSAUR	A Epic ... 96B	2006	English	Not Applicable	6	0.99
1002	2006-02-15 05:03:42	ACE GOLDFINGER	A Astou... 100B	2006	English	Not Applicable	3	4.99
1003	2006-02-15 05:03:42	ADAPTATION HOLES	A Astou... 96B	2006	English	Not Applicable	7	2.99
1004	2006-02-15 05:03:42	AFFAIR PREJUDICE	A Fanci... 92B	2006	English	Not Applicable	5	2.99
1005	2006-02-15 05:03:42	AFRICAN EGG	A Fast-... 117B	2006	English	Not Applicable	6	2.99
1006	2006-02-15 05:03:42	AGENT TRUMAN	A Intre... 89B	2006	English	Not Applicable	3	2.99
1007	2006-02-15 05:03:42	AIRPLANE SIERRA	A Touch... 81B	2006	English	Not Applicable	6	4.99
1008	2006-02-15 05:03:42	AIRPORT POLLOCK	A Epic ... 77B	2006	English	Not Applicable	6	4.99
1009	2006-02-15 05:03:42	ALABAMA DEVIL	A Thoug... 115B	2006	English	Not Applicable	3	2.99
1010	2006-02-15 05:03:42	ALADDIN CALENDAR	A Actio... 89B	2006	English	Not Applicable	6	4.99

图 8-180　维度表 dim_film

从图 8-181 中可以看出，维度表 dim_film 和 dim_film_actor_bridge 中均插入了数据，说明我们成功实现了加载电影数据至电影维度表 dim_film 中。

8.3.9　加载租赁数据至租赁事实表

下面通过 Kettle 工具加载租赁数据至租赁事实表 fact_rental，具体实现步骤如下。

film_key	actor_key	actor_weighting_factor
1001	210	0.20
1001	230	0.20
1001	253	0.20
1001	362	0.20
1001	398	0.20
1002	285	0.50
1002	360	0.50
1003	219	0.33
1003	264	0.33

图 8-181 维度表 dim_film_actor_bridge

1. 打开 Kettle 工具，创建转换

使用 Kettle 工具创建转换 load_fact_rental，并添加“表输入”控件、“字段选择”控件、“过滤记录”控件、“计算器”控件、“增加常量”控件、“数据库查询”控件、“维度查询/更新”控件、“插入/更新”控件以及 Hop 跳连接线，具体效果如图 8-182 所示。

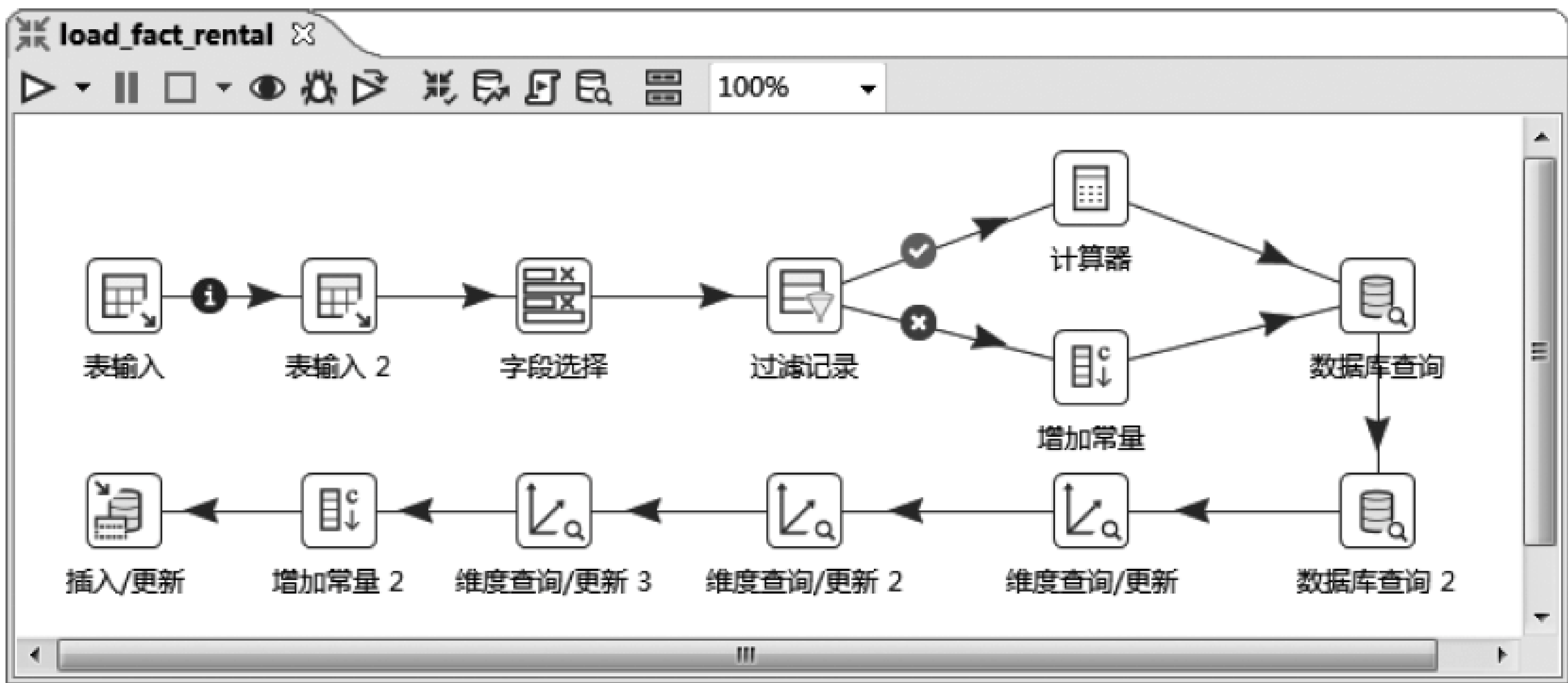


图 8-182 创建转换 load_fact_rental

2. 配置“表输入”控件

双击图 8-182 中的“表输入”控件，进入“表输入”界面，如图 8-183 所示。

表输入
步骤名称 表输入
数据库连接 编辑... 新建... Wizard...
SQL 获取SQL查询语句...
SELECT <values> FROM <table name> WHERE <conditions>
行1 列0
允许简易转换
替换 SQL 语句里的变量
从步骤插入数据
执行每一行?
记录数量限制 0
Help 确定(O) 预览(P) 取消(C)

图 8-183 “表输入”界面

在图 8-183 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-184 所示。



图 8-184 MySQL 数据库连接的配置

在图 8-183 的 SQL 框中编写 SQL 语句，用于获取字段 rental_last_update 中的最大值，将该值替换为 1970-01-01 00:00:00 并赋值给临时字段 max_fact_rental_last_update；单击“预览”按钮，查看临时字段 max_fact_rental_last_update 是否将默认值设置为 1970-01-01 00:00:00，具体如图 8-185 和图 8-186 所示。



图 8-185 编写 SQL 语句

从图 8-186 中可以看出，临时字段 max_fact_rental_last_update 的默认值设置为 1970-01-01 00:00:00，单击“关闭”→“确定”按钮，完成“表输入”控件的配置。



图 8-186　预览数据

3. 配置“表输入 2”控件

双击图 8-182 中的“表输入 2”控件，进入“表输入”界面，如图 8-187 所示。



图 8-187　“表输入”界面

在图 8-187 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-188 所示。

在图 8-187 的 SQL 框中编写 SQL 语句，用于获取数据库 sakila 中数据表 rental 中的最新数据，具体如图 8-189 所示。

在图 8-189 中单击“确定”按钮，完成“表输入 2”控件的配置。

4. 配置“字段选择”控件

双击图 8-182 中的“字段选择”控件，进入“选择/改名值”界面，在“选择和修改”选项卡中添加要修改的字段，如图 8-190 所示；在“元数据”选项卡中的“需要改变元数据的字段”处添加字段。这里使用“字段选择”控件构建数据仓库中维度表需要的字段数据，其配置如图 8-191 所示。

在图 8-191 中单击“确定”按钮，完成“字段选择”控件的配置。

5. 配置“过滤记录”控件

双击图 8-182 中的“过滤记录”控件，进入“过滤记录”界面，如图 8-192 所示。

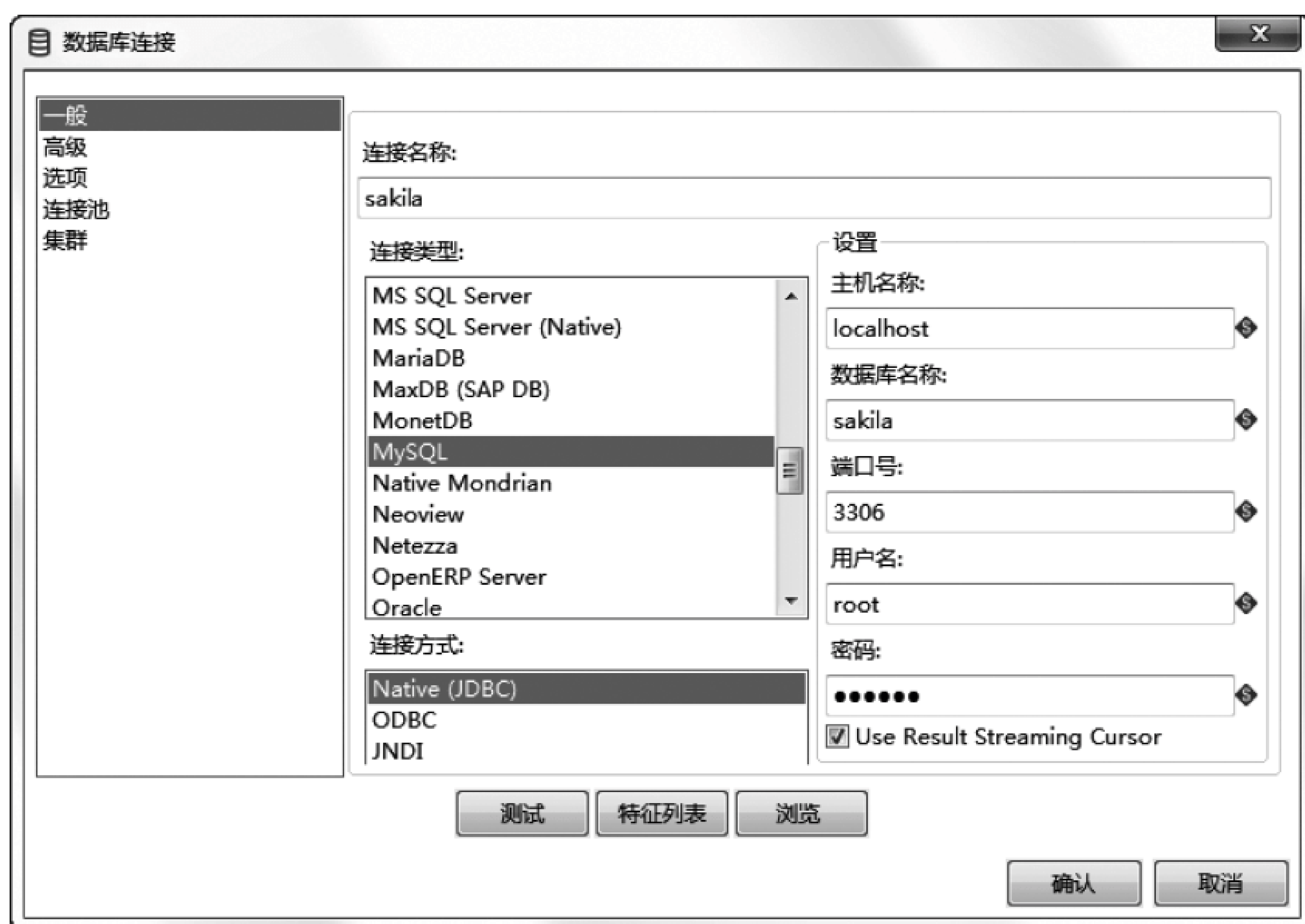


图 8-188 MySQL 数据库连接的配置



图 8-189 编写 SQL 语句

在图 8-192 中的“条件”处设置过滤的条件，对归还时间进行过滤操作；单击＜field＞框，弹出“字段”对话框，选择要过滤的字段 return_datetime(归还时间)，如图 8-193 所示。

在图 8-193 中单击“确定”按钮，完成过滤字段 return_datetime 的选择。

单击图 8-192 中的“=”框，弹出“函数:”对话框，选择过滤条件(这里选择的是 IS NOT NULL(不为空))，如图 8-194 所示。

在图 8-194 中单击“确定”按钮，完成过滤条件的选择，判断归还时间不为空。字段



图 8-190　添加要修改的字段



图 8-191　添加要修改的元数据



图 8-192　“过滤记录”界面

return_datetime 的过滤设置如图 8-195 所示。

在图 8-195 中“发送 true 数据给步骤：”后的下拉列表中选择“计算器”，将字段 return_datetime 不为空的数据传递到计算器控件流中；在“发送 false 数据给步骤：”后的下拉列表中选择“增加常量”，将为空的数据传递到“增加常量”控件流中，具体如图 8-196 所示。

在图 8-196 中单击“确定”按钮，完成“过滤记录”控件的配置。



图 8-193 “字段”对话框



图 8-194 “函数:”对话框



图 8-195 字段 return_datetime 的过滤设置



图 8-196 配置“过滤记录”控件

6. 配置“计算器”控件

双击图 8-182 中的“计算器”控件,进入“计算器”界面,在“字段”处添加新字段 milisecs、rental_duration_milisecs、rental_duration、count_returns、return_date_key1,其中字段 milisecs 为自定义常量,值为 1000;字段 rental_duration_milisecs 用于存储租赁的毫秒数;

字段 rental_duration 用于存储租赁的周期；字段 count_returns 为自定义字段，用于统计归还的次数；字段 return_date_key1 用于存储归还的日期。这里使用“计算器”控件计算租赁的周期，其配置如图 8-197 所示。

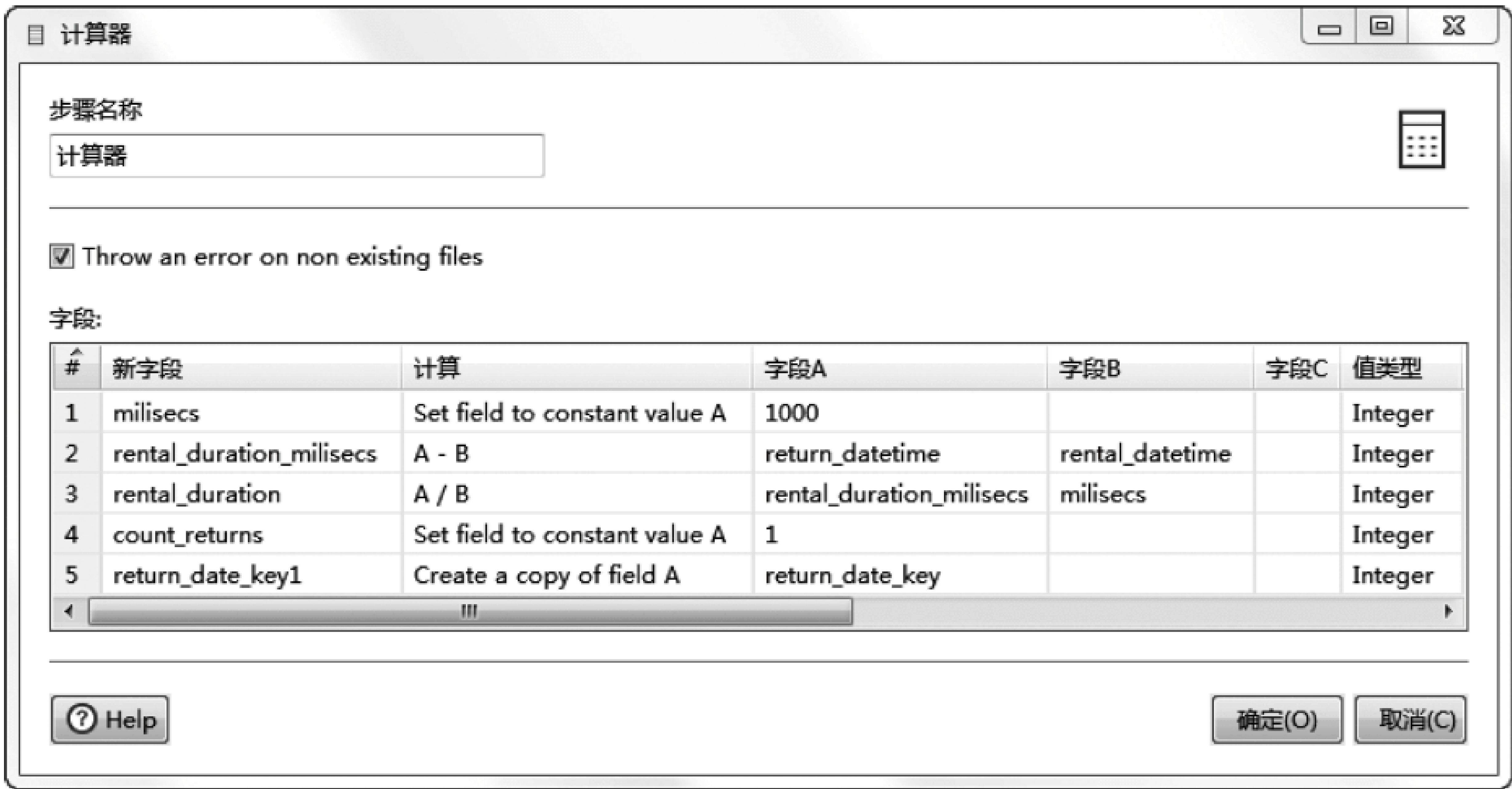


图 8-197　配置“计算器”控件

7. 配置“增加常量”控件

双击图 8-182 中的“增加常量”控件，进入“增加常量”界面，在“字段”框中添加常量字段 rental_duration、count_returns、return_date_key1，用来记录归还的日期(由于在“过滤记录”控件中将归还日期为空的字段输出到“增加常量”控件流中，因此需要在“增加常量”控件中添加用于记录归还日期的字段)，具体如图 8-198 所示。



图 8-198　配置“增加常量”控件

在图 8-198 中单击“确定”按钮，完成“增加常量”控件的配置。

8. 配置“数据库查询”控件

双击图 8-182 中的“数据库查询”控件，进入“数据库查询”界面，如图 8-199 所示。

在图 8-199 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-200 所示。



图 8-199 “数据库查询”界面



图 8-200 MySQL 数据库连接的配置

单击图 8-199 中表名右侧的“浏览”按钮，添加数据表 inventory，用于查询电影库存的信息；在“查询所需的关键字”框中添加查询所需的关键字字段 inventory_id，用于指定表字段和流字段的比较条件；在“查询表返回的值”框中添加查询表返回的值，即字段 film_id 和 store_id 的数据，如图 8-201 所示。

在图 8-201 中单击“确定”按钮，完成“数据库查询”控件的配置。



图 8-201　配置"数据库查询"控件

9. 配置"数据库查询 2"控件

双击图 8-182 中的"数据库查询 2"控件，进入"数据库查询"界面，如图 8-202 所示。



图 8-202　"数据库查询"界面

在图 8-202 中单击"新建"按钮，配置数据库连接，配置完成后单击"确认"按钮。MySQL 数据库连接的配置如图 8-203 所示。



图 8-203　MySQL 数据库连接的配置

单击图 8-202 中表名右侧的"浏览"按钮，添加维度表 dim_film，用于获取数据仓库中维度表 dim_film 中的数据；在"查询所需的关键字"框中添加查询所需的关键字字段 film_id，用于指定表字段和流字段的比较条件；在"查询表返回的值"框中添加查询表返回的值，即字段 film_key，如图 8-204 所示。



图 8-204　配置"数据库查询 2"控件

在图 8-204 中单击“确定”按钮，完成“数据库查询 2”控件的配置。

10. 配置“维度查询/更新”控件

双击图 8-182 中的“维度查询/更新”控件，进入“维度查询/更新”界面，具体如图 8-205 所示。

图 8-205 “维度查询/更新”界面

在图 8-205 中单击“新建”按钮，配置数据库连接，完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-206 所示。

在图 8-205 中单击目标表右侧的“浏览”按钮，选择输出的目标表，即维度表 dim_customer；在“关键字”选项卡处添加关键字字段；在“代理关键字段”后的下拉列表中选择 customer_key，并指定“创建代理键”为使用自增字段；在“Version 字段”后的下拉列表中选择 customer_version_number；在“Stream 日期字段”后的下拉列表中选择 rental_datetime；在“开始日期字段”后的下拉列表中选择 customer_valid_from；在“截止日期字段”后的下拉列表中选择 customer_valid_through，具体如图 8-207 所示。

在图 8-207 中单击“确定”按钮，完成“维度查询/更新”控件的配置。



图 8-206 MySQL 数据库连接的配置

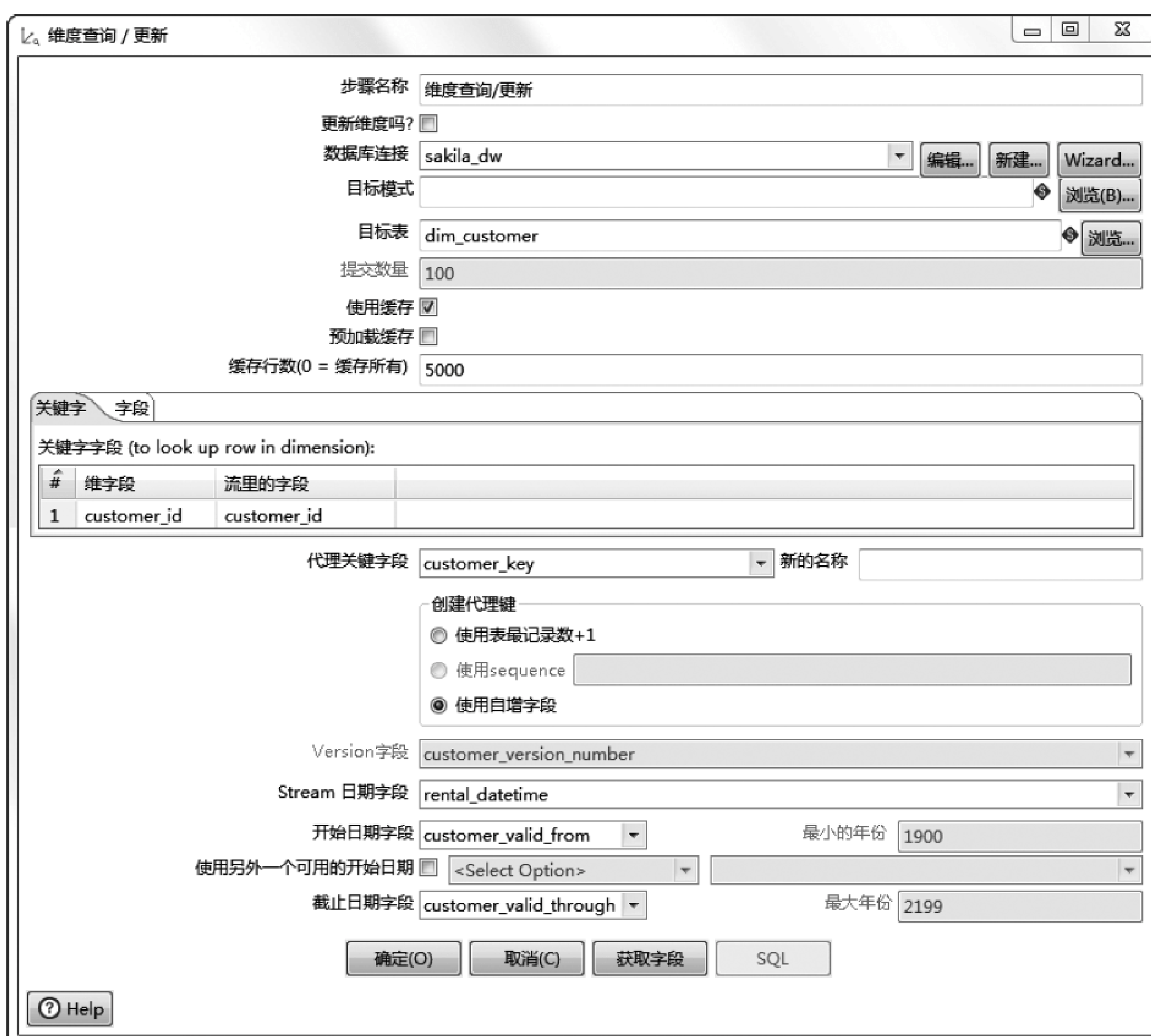


图 8-207 配置“维度查询/更新”控件

11. 配置“维度查询/更新 2”控件

双击图 8-182 中的“维度查询/更新 2”控件,进入“维度查询/更新”界面,具体如图 8-208 所示。

图 8-208 “维度查询/更新”界面

在图 8-208 中单击“新建”按钮,配置数据库连接,配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-209 所示。

在图 8-208 中单击目标表右侧的“浏览”按钮,选择输出的目标表,即维度表 dim_staff;在“关键字”选项卡中添加关键字字段 staff_id;在“代理关键字段”后的下拉列表中选择 staff_key,并指定“创建代理键”是使用自增字段;在“Version 字段”后的下拉列表中选择 staff_version_number;在“Stream 日期字段”后的下拉列表中选择 rental_datetime;在“开始日期字段”后的下拉列表中选择 staff_valid_from;在“截止日期字段”后的下拉列表中选择 staff_valid_through,具体如图 8-210 所示。



图 8-209　MySQL 数据库连接的配置

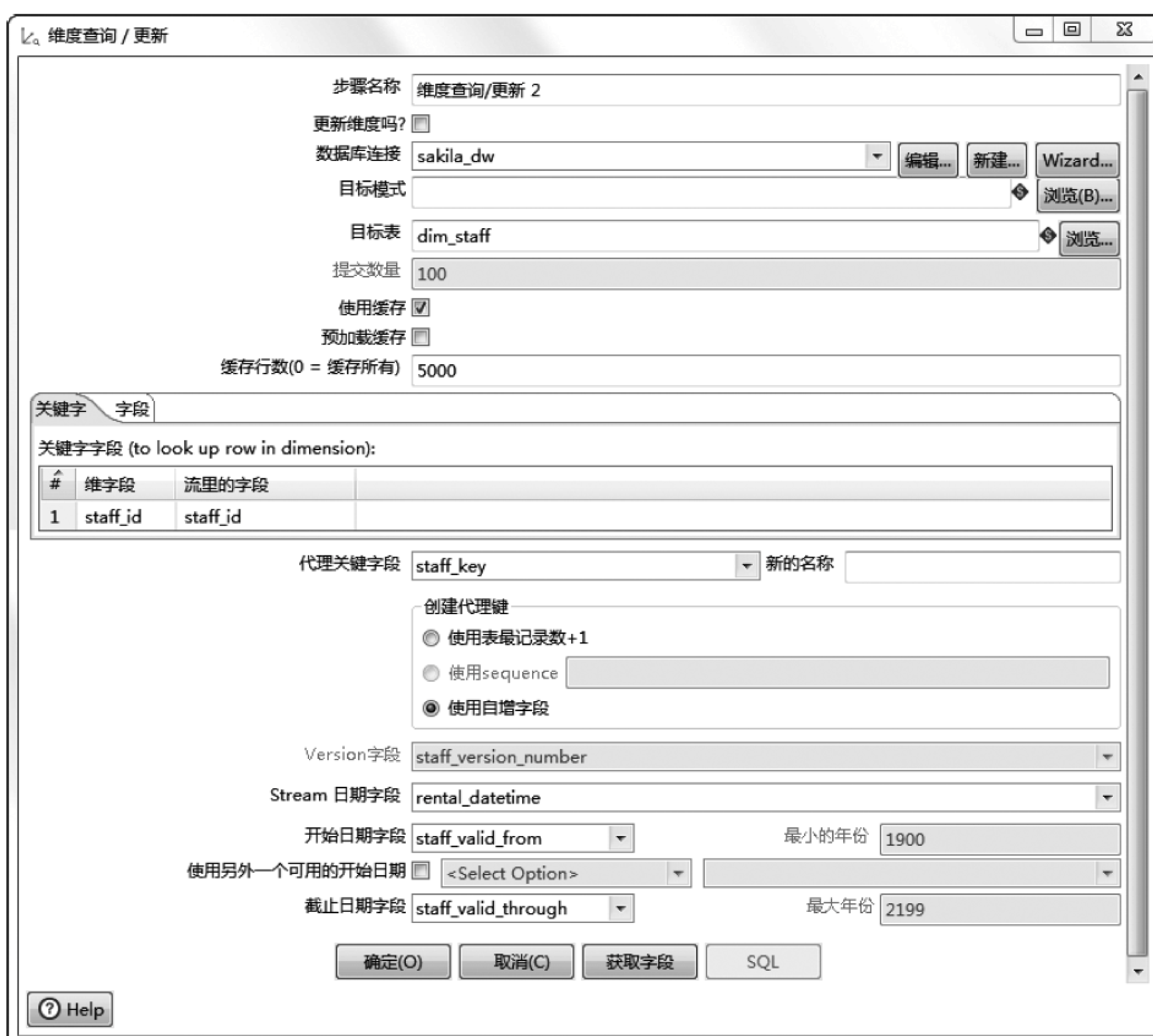


图 8-210　配置“维度查询/更新 2”控件

在图 8-210 中单击“确定”按钮，完成“维度查询/更新 2”控件的配置。

12. 配置“维度查询/更新 3”控件

双击图 8-182 中的“维度查询/更新 3”控件，进入“维度查询/更新”界面，具体如图 8-211 所示。

图 8-211　“维度查询/更新”界面

在图 8-211 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-212 所示。

在图 8-211 中单击目标表右侧的“浏览”按钮，选择输出的目标表，即维度表 dim_store；在“关键字”选项卡中添加关键字字段 store_id；在“代理关键字段”后的下拉列表中选择 store_key，并指定“创建代理键”是使用自增字段；在“Version 字段”后的下拉列表中选择 store_version_number；在“Stream 日期字段”后的下拉列表中选择 rental_datetime；在“开始日期字段”后的下拉列表中选择 store_valid_from；在“截止日期字段”后的下拉列表中选择 store_valid_through，具体如图 8-213 所示。

在图 8-213 中单击“确定”按钮，完成“维度查询/更新 3”控件的配置。



图 8-212 MySQL 数据库连接的配置

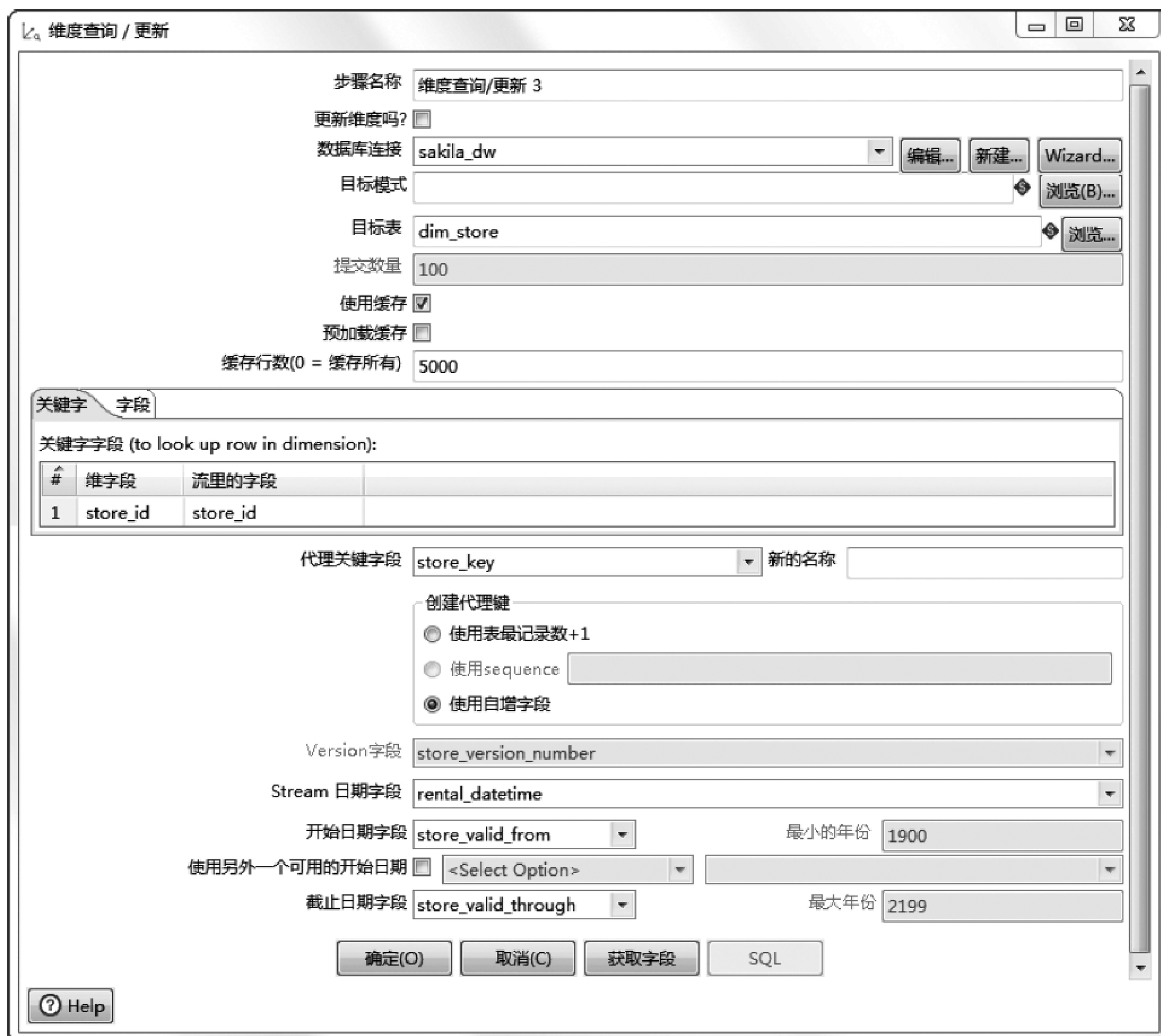


图 8-213 配置"维度查询/更新 3"控件

13. 配置“增加常量 2”控件

双击图 8-182 中的“增加常量 2”控件，进入“增加常量”界面，在“字段”框中添加常量字段 count_rentals，用于统计租赁的次数，具体如图 8-214 所示。



图 8-214　配置“增加常量 2”控件

在图 8-214 中单击“确定”按钮，完成“增加常量 2”控件的配置。

14. 配置“插入/更新”控件

双击图 8-182 中的“插入/更新”控件，进入“插入/更新”界面，如图 8-215 所示。



图 8-215　“插入/更新”界面

在图 8-215 中单击“新建”按钮，配置数据库连接，配置完成后单击“确认”按钮。MySQL 数据库连接的配置如图 8-216 所示。

单击图 8-215 中目标表右侧的“浏览”按钮，弹出“数据库浏览器”窗口，选择目标表，即事实表 fact_rental；单击“获取字段”按钮，用来指定查询数据需要的关键字字段 rental_id，用于指定表字段和流字段的比较条件；单击“获取和更新字段”按钮，用来指定需要更新的字段，具体如图 8-217 所示。

在图 8-217 中单击“确定”按钮，完成“插入/更新”控件的配置。



图 8-216 MySQL 数据库连接的配置



图 8-217 配置“插入/更新”控件

15. 运行转换 load_fact_rental

单击转换工作区顶部的▷按钮,运行创建的转换 load_fact_rental,实现加载租赁数据至租赁事实表 fact_rental 中,具体如图 8-218 所示。

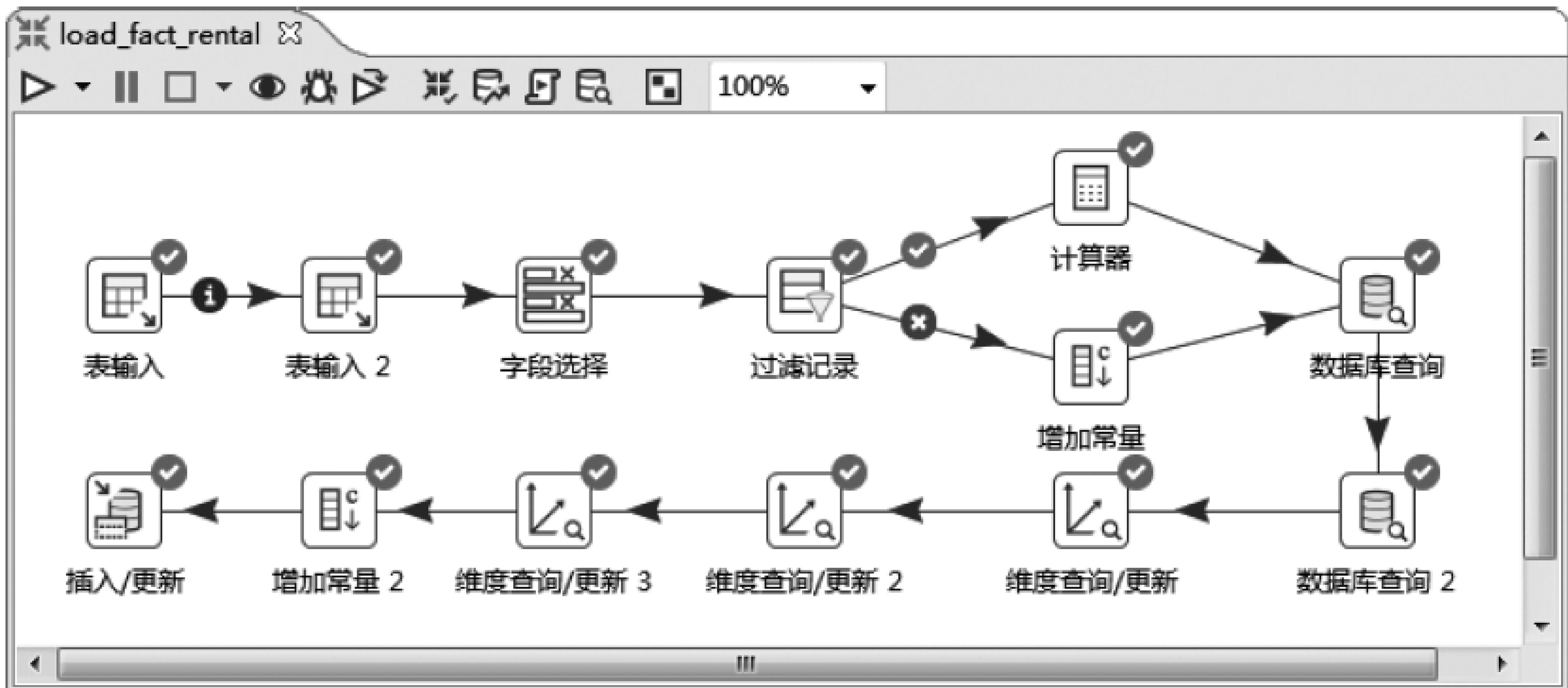


图 8-218 运行转换 load_fact_rental

从图 8-218 中的执行结果看,每个控件的右上角均有"√",说明转换 load_fact_rental 执行成功。

16. 查看事实表 fact_rental 中的数据

通过 SQLyog 工具,查看事实表 fact_rental 是否已成功插入数据,查看结果如图 8-219 所示(这里只展示部分数据)。

cust...	s..	film_key	st...	renta...	return_d...	rent...	co...	c..	rental...	rental_last_update
2527	9	1080	4	20050524	20050526	225330	1	1	169860	2006-02-15 21:30:53
2856	9	1333	5	20050524	20050528	225433	1	1	333960	2006-02-15 21:30:53
2805	9	1373	5	20050524	20050601	230339	1	1	688140	2006-02-15 21:30:53
2730	10	1535	4	20050524	20050603	230441	1	1	787140	2006-02-15 21:30:53
2619	9	1450	5	20050524	20050602	230521	1	1	710880	2006-02-15 21:30:53
2946	9	1613	4	20050524	20050527	230807	1	1	181440	2006-02-15 21:30:53
2666	10	1870	5	20050524	20050529	231153	1	1	422580	2006-02-15 21:30:53
2636	10	1510	4	20050524	20050527	233146	1	1	259320	2006-02-15 21:30:53
2523	9	1565	4	20050525	20050528	40	1	1	260520	2006-02-15 21:30:53

图 8-219 事实表 fact_rental

从图 8-219 中可以看出,事实表 fact_rental 中已插入数据,说明我们成功实现了加载租赁数据至租赁事实表 fact_rental 中。

8.3.10 加载数据库 sakila 中的数据至数据仓库 sakila_dw

下面通过 Kettle 工具将前面创建的转换整合成一个整体,定时加载 sakila 中的数据至数据仓库 sakila_dw 中,具体实现步骤如下。

1. 打开 Kettle 工具,创建作业

使用 Kettle 工具创建作业 load_rentals,并添加 Start 控件、"转换"控件、"发送邮件"控

件、“中止作业”控件以及 Hop 作业项连接线，具体效果如图 8-220 所示。

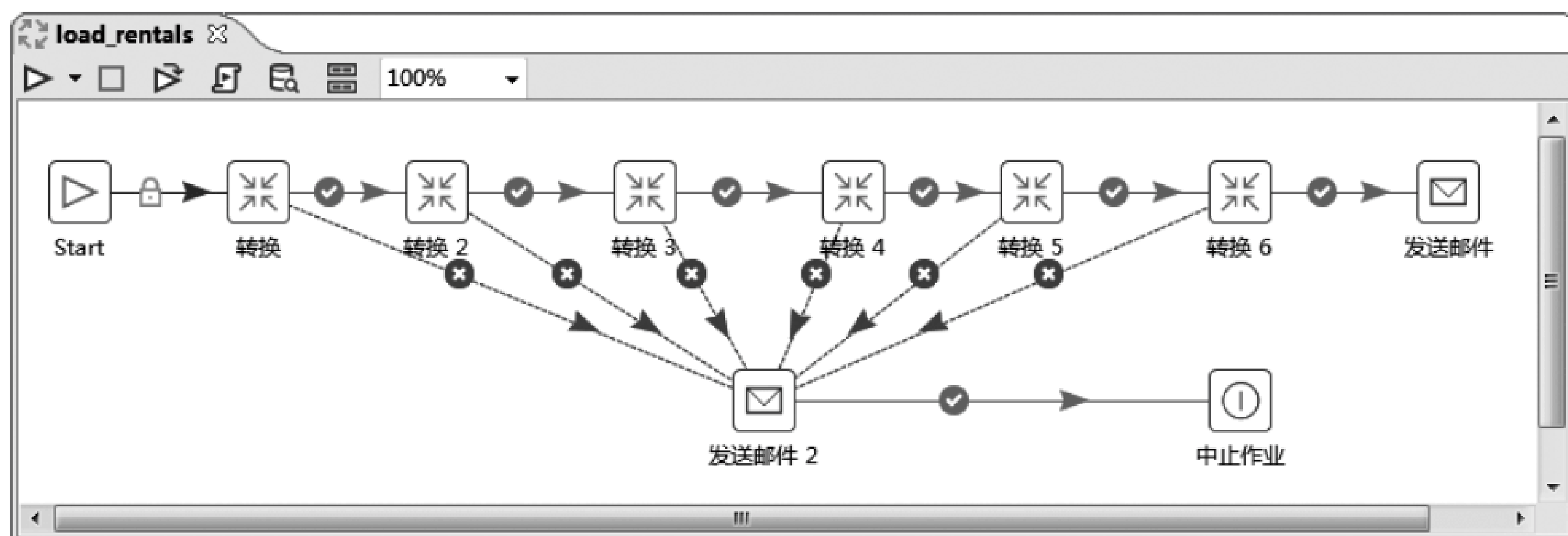


图 8-220 创建作业 load_rentals

2. 配置“转换”控件

双击图 8-220 中的“转换”控件，进入“转换”界面，单击 Transformation 后的“浏览”按钮，选择添加转换 load_dim_staff，具体如图 8-221 所示。

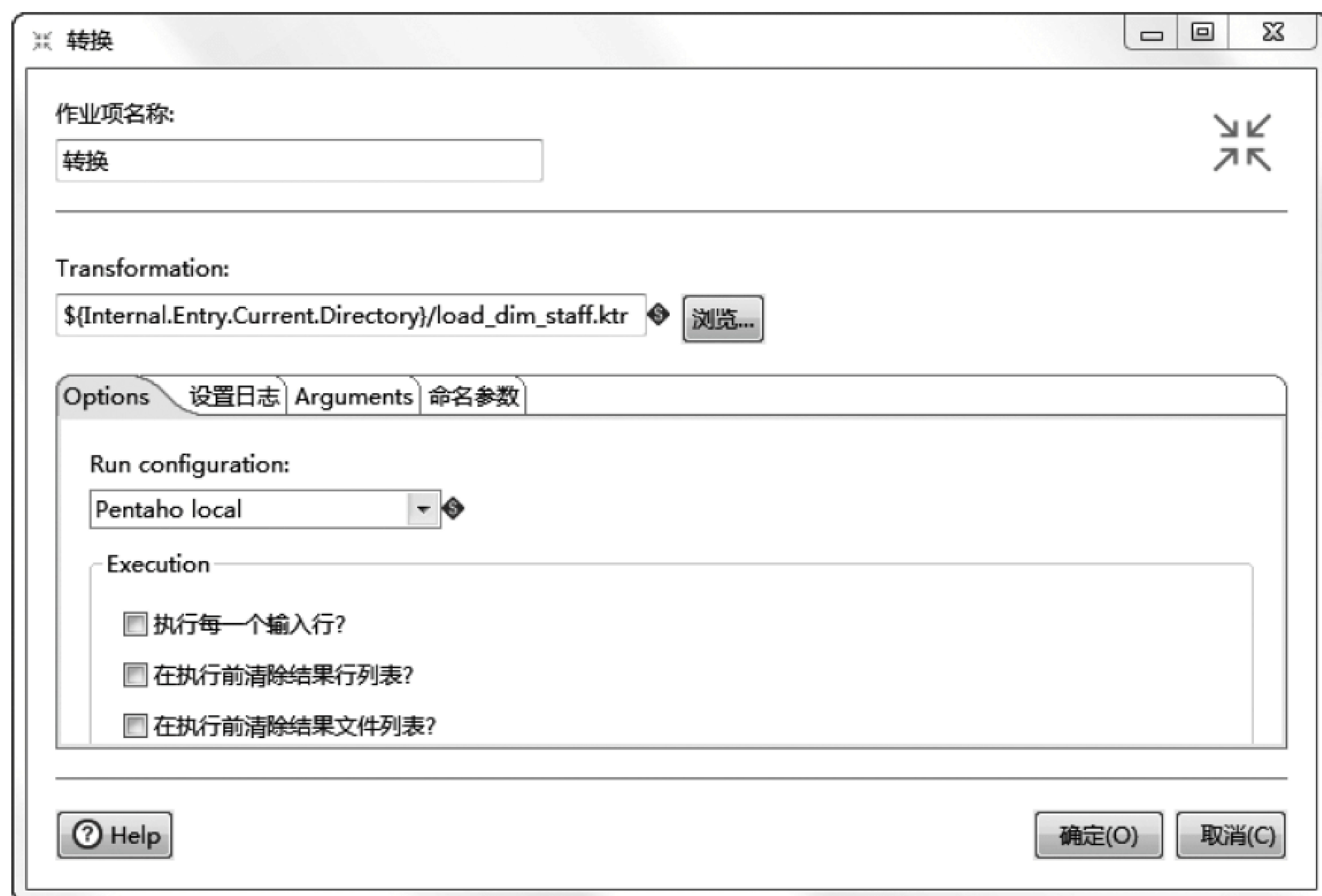


图 8-221 配置“转换”控件

在图 8-221 中单击“确定”按钮，完成“转换”控件的配置。

3. 配置“转换 2”控件

双击图 8-220 中的“转换 2”控件，进入“转换”界面，单击 Transformation 后的“浏览”按钮，选择添加转换 load_dim_customer，具体如图 8-222 所示。

在图 8-222 中单击“确定”按钮，完成“转换 2”控件的配置。

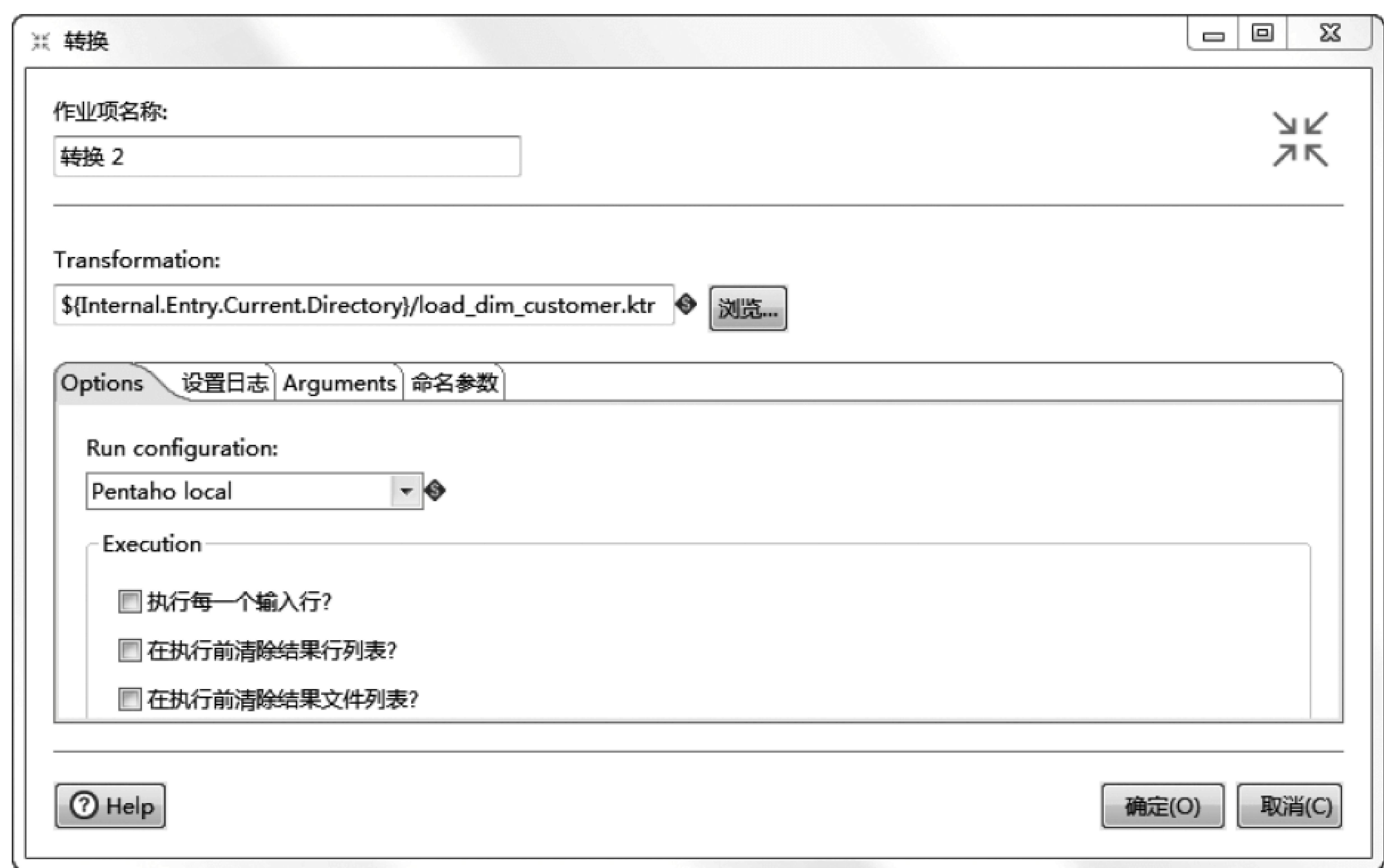


图 8-222 配置“转换 2”控件

4. 配置“转换 3”控件

双击图 8-220 中的“转换 3”控件，进入“转换”界面，单击 Transformation 后的“浏览”按钮，选择添加转换 load_dim_store，具体如图 8-223 所示。



图 8-223 配置“转换 3”控件

在图 8-223 中单击“确定”按钮，完成“转换 3”控件的配置。

5. 配置"转换 4"控件

双击图 8-220 中的"转换 4"控件,进入"转换"界面,单击 Transformation 处的"浏览"按钮,选择添加转换 load_dim_actor,具体如图 8-224 所示。



图 8-224　配置"转换 4"控件

在图 8-224 中单击"确定"按钮,完成"转换 4"控件的配置。

6. 配置"转换 5"控件

双击图 8-220 中的"转换 5"控件,进入"转换"界面,单击 Transformation 后的"浏览"按钮,选择添加转换 load_dim_film,具体如图 8-225 所示。



图 8-225　配置"转换 5"控件

在图8-225中单击“确定”按钮，完成“转换5”控件的配置。

7. 配置“转换6”控件

双击图8-220中的“转换6”控件，进入“转换”界面，单击Transformation处的“浏览”按钮，选择添加转换load_fact_rental，具体如图8-226所示。



图8-226　配置“转换6”控件

在图8-226中单击“确定”按钮，完成“转换6”控件的配置。

8. 配置“发送邮件”控件

双击图8-220中的“发送邮件”控件，进入“发送邮件”界面，在“地址”选项卡中添加收件人和发件人的信息，如图8-227所示；在“服务器”选项卡中添加邮件服务器和验证的信息，



图8-227　添加收件人和发件人

如图 8-228 所示;在“邮件消息”选项卡中添加消息内容,如图 8-229 所示。这里使用“发送邮件”控件,主要用于接收作业运行成功的提醒。



图 8-228 添加邮件服务器和验证信息



图 8-229 添加消息内容

在图 8-229 中单击“确定”按钮,完成“发送邮件”控件的配置。

9. 配置“发送邮件 2”控件

双击图 8-220 中的“发送邮件 2”控件,进入“发送邮件”界面,在“地址”选项卡中添加收件人和发件人的信息,如图 8-230 所示;在“服务器”选项卡中添加邮件服务器和验证的信息,如图 8-231 所示;在“邮件消息”选项卡中添加消息内容,如图 8-232 所示。“发送邮件 2”

控件主要用于接收作业运行错误的提醒信息。



图 8-230　添加收件人和发件人



图 8-231　添加邮件服务器和验证信息

在图 8-232 中单击“确定”按钮，完成“发送邮件 2”控件的配置。

10．运行作业 load_rentals

单击作业工作区顶部的▷按钮，运行创建的作业 load_ rentals，实现加载数据库 sakila 中的数据至数据仓库 sakila_dw 中，具体如图 8-233 所示。

从图 8-233 中的执行结果看，每个控件的右上角均有“√”，说明作业 load_rentals 执行成功。

11．查看数据仓库 sakila_dw 中维度表的数据

通过 SQLyog 工具，查看数据仓库 sakila_dw 中的维度表和事实表是否已成功插入数据，查看结果如图 8-234～图 8-241 所示。



图 8-232 添加消息内容

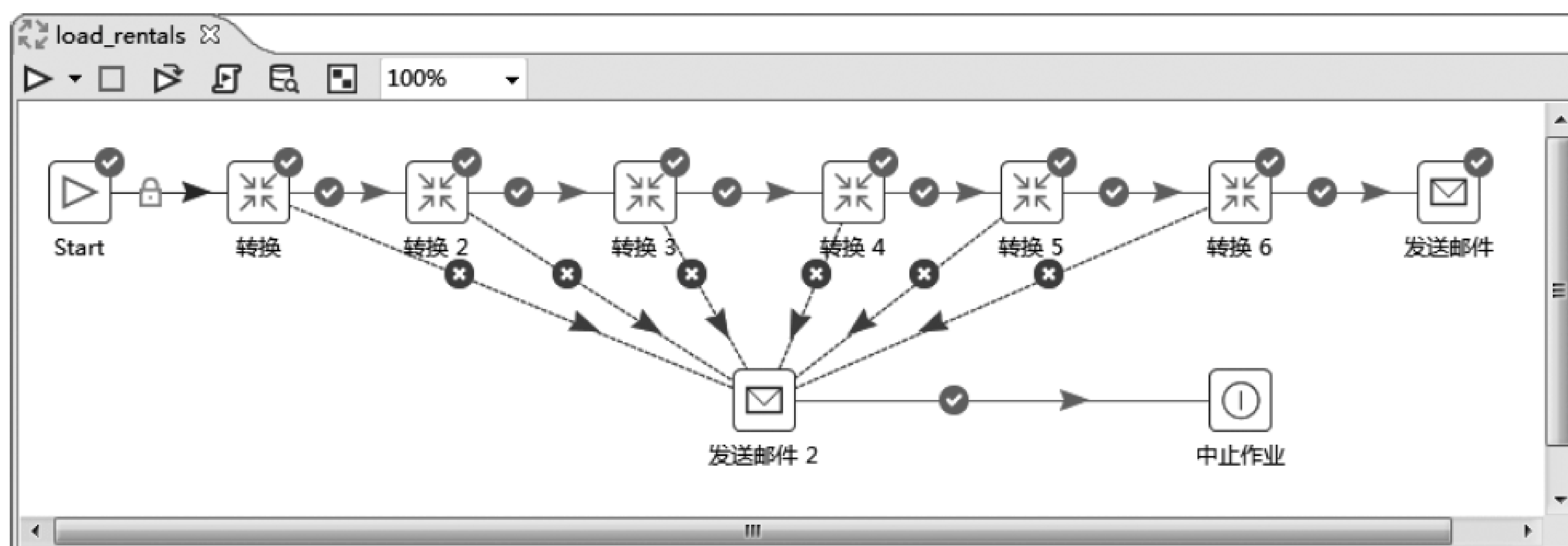


图 8-233 运行作业 load_ rentals

actor_key	actor_last_update	actor_last_name	actor_first_name	actor_id
401	2006-02-15 04:34:33	GUINESS	PENELOPE	1
402	2006-02-15 04:34:33	WAHLBERG	NICK	2
403	2006-02-15 04:34:33	CHASE	ED	3
404	2006-02-15 04:34:33	DAVIS	JENNIFER	4
405	2006-02-15 04:34:33	LOLLOBRIGIDA	JOHNNY	5
406	2006-02-15 04:34:33	NICHOLSON	BETTE	6
407	2006-02-15 04:34:33	MOSTEL	GRACE	7
408	2006-02-15 04:34:33	JOHANSSON	MATTHEW	8
409	2006-02-15 04:34:33	SWANK	JOE	9
410	2006-02-15 04:34:33	GABLE	CHRISTIAN	10

图 8-234 维度表 dim_actor 中的数据

从上述图中可以看出，数据仓库 sakila_dw 中的维度表和事实表均插入了数据，说明我们通过 Kettle 工具实现了将数据库 sakila 中的数据加载到数据仓库 sakila_dw 中。

cust...	customer_last_up...	custom...	custo...	customer_la...	customer...	customer
4791	2006-02-15 04:57:20	597	FREDDIE	DUGGAN	FREDDIE.DUGG	Yes
4792	2006-02-15 04:57:20	598	WADE	DELVALLE	WADE.DELVALL	Yes
4793	2006-02-15 04:57:20	599	AUSTIN	CINTRON	AUSTIN.CINTR	Yes
4790	2006-02-15 04:57:20	596	ENRIQUE	FORSYTHE	ENRIQUE.FORS	Yes
4789	2006-02-15 04:57:20	595	TERRENCE	GUNDERSON	TERRENCE.GUN	Yes
4786	2006-02-15 04:57:20	592	TERRANCE	ROUSH	TERRANCE.ROU	No
4787	2006-02-15 04:57:20	593	RENE	MCALISTER	RENE.MCALIST	Yes
4788	2006-02-15 04:57:20	594	EDUARDO	HIATT	EDUARDO.HIAT	Yes
4785	2006-02-15 04:57:20	591	KENT	ARSENAULT	KENT.ARSENAU	Yes

图 8-235　维度表 dim_customer 中的数据

date_key	date_v...	date_short	date_medium	date_long	date_full
20091228	2009-12-28	12/28/09	Dec 28, 2009	December 28, 2009	Monday, December 28, 20(
20091227	2009-12-27	12/27/09	Dec 27, 2009	December 27, 2009	Sunday, December 27, 20(
20091226	2009-12-26	12/26/09	Dec 26, 2009	December 26, 2009	Saturday, December 26, :
20091225	2009-12-25	12/25/09	Dec 25, 2009	December 25, 2009	Friday, December 25, 20(
20091224	2009-12-24	12/24/09	Dec 24, 2009	December 24, 2009	Thursday, December 24, :
20091223	2009-12-23	12/23/09	Dec 23, 2009	December 23, 2009	Wednesday, December 23,
20091222	2009-12-22	12/22/09	Dec 22, 2009	December 22, 2009	Tuesday, December 22, 2(
20091221	2009-12-21	12/21/09	Dec 21, 2009	December 21, 2009	Monday, December 21, 20(
20091220	2009-12-20	12/20/09	Dec 20, 2009	December 20, 2009	Sunday, December 20, 20(

图 8-236　维度表 dim_date 中的数据

film_key	film_last_update	film_title	film_description		fil...	film...	film
2907	2006-02-15 05:03:42	TRANSLATION SUMMI	A Touch...	93B	2006	English	Not A
2906	2006-02-15 05:03:42	TRAMP OTHERS	A Brill...	87B	2006	English	Not A
2905	2006-02-15 05:03:42	TRAINSPOTTING STI	A Fast-...	90B	2006	English	Not A
2904	2006-02-15 05:03:42	TRAIN BUNCH	A Thril...	90B	2006	English	Not A
2903	2006-02-15 05:03:42	TRAFFIC HOBBIT	A Amazi...	102B	2006	English	Not A
2902	2006-02-15 05:03:42	TRADING PINOCCHI(	A Emoti...	117B	2006	English	Not A
2901	2006-02-15 05:03:42	TRACY CIDER	A Touch...	101B	2006	English	Not A
2900	2006-02-15 05:03:42	TOWN ARK	A Awe-I...	98B	2006	English	Not A
2899	2006-02-15 05:03:42	TOWERS HURRICANE	A Fatef...	85B	2006	English	Not A

图 8-237　维度表 dim_film 中的数据

film_key	actor_key	actor_weighting_factor
1001	210	0.20
1001	230	0.20
1001	253	0.20
1001	362	0.20
1001	398	0.20
1002	285	0.50
1002	360	0.50
1003	219	0.33
1003	264	0.33
1004	241	0.50

图 8-238　维度表 dim_film_actor_bridge 中的数据

staff_key	staff_last_u...	staff_f...	staff_...	sta...	staff_...	staff_v...	staff_...
15	1970-01-01 00:00	Jon	Stephens	2	2	1	1900-01-
16	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	1	(NULL)
14	1970-01-01 00:00	Mike	Hillyer	1	1	1	1900-01-
13	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	1	(NULL)
(Auto)	1970-01-01 00:00	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

图 8-239　维度表 dim_staff 中的数据

time_key	time_value	hours24	hours12	minutes	seconds	am_pm
0	00:00:00	0	0	0	0	AM
1	00:00:01	0	0	0	1	AM
2	00:00:02	0	0	0	2	AM
3	00:00:03	0	0	0	3	AM

图 8-240 维度表 dim_time 中的数据

cust...	s..	film_key	s...	renta...	return_d...	rent...	co...	c..	ren...	rental_las
4748	15	2001	7	20050821	0	3032	0	1	(NULL)	2006-02-23
2908	10	1135	5	20050529	20050602	210032	1	1	324900	2006-02-15
2419	10	1567	4	20050529	20050607	210722	1	1	702900	2006-02-15
2745	9	1741	5	20050529	20050604	213112	1	1	523080	2006-02-15

图 8-241 事实表 fact_rental 中的数据

8.4 本章小结

本章主要讲解了构建 DVD 租赁商店数据仓库，并将数据库 sakila 中的数据加载至数据仓库 sakila_dw 中的相关知识，包括案例概述、数据准备以及案例实现。希望读者通过本章的学习，可以掌握构建数据仓库以及将数据库中的数据进行相关的清洗操作，最终加载至数据仓库中，便于后续在线 DVD 租赁商店的决策者对数据进行分析得出商业决策。